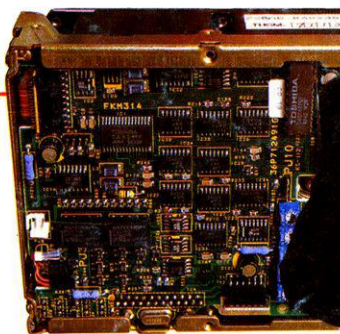
Anno IX - N. 79 - 25 DICEMBRE 1990 - Sped. Abb. Post. Gr III/70 - CR - Distr.: Parrini

Il modem parla italiano

MUSICA musica

AMIGA
TUTTO SULL' HARD DISK

COME TRASFORMARE L'AMIGA 500 IN UN VERO PC

L'amiga del mistero

Postamiga • Sfida ai lettori • A scuola con Microsoft • Videogames, le novità • I linguaggi del mondo Ms-Dos: programmazione in Turbo Pascal, Quick Basic, Turbo C • Il Dos del PC • Come fondere l.m. e Basic con il C/64 • Schede grafiche

Leggo VR perché mi dà la rotta

Sommario

Amigames

Da pagina **43** a pagina **49** la consueta rassegna dei videogames in arrivo; e i relativi commenti, severi come al solito!

Campus 64 / 128

32 Stampa in quattro toni di grigio (Software)
36 Una fusione tra Basic e Linguaggio Macchina (Linker per C64)
81 Un domino per Amiga e C64

Campus Amiga

65 Messaggi cifrati per Amiga
71 Amigafacile, i comandi del Dos
77 Amiga in musica
81 Un domino per Amiga e C64
86 PostAmiga, i dubbi dei lettori

Usa il tuo computer

12 All'inizio era il **Dos** (Insieme)
18 Ed è subito prospettiva (**Ms-Dos**, linguaggi)
25 A scuola con **Microsoft** (insieme)
28 Superscheda Supergrafica (Hardware Ms-Dos)
39 Un file servitore di tre padroni (**Ms-Dos**, linguaggio "**C**")
50 Hardware per **Ms-Dos** e **Amiga**
91 Pronto, chi **Modem**?

Rubriche

4 Chi troppo vuole...
5 La vostra posta
8 Systems per te
62 Digitando...

E' bene ricordare che...

Parleremo del **C/128** solo fino alla fine del 1990.
Dal **gennaio '91** verrà evasa prevalentemente la corrispondenza pervenuta a mezzo BBS (modem).
Dal **gennaio '91** verranno privilegiate le collaborazioni che perverranno in Redazione a mezzo BBS.

Tra breve verrà dato ampio spazio, oltre che ad **Amiga**, anche al favoloso mondo **Ms-Dos**! (ma già lo stiamo facendo...)
Dal **dicembre '91** non verranno più affrontati argomenti relativi al C/64.
...è doveroso augurarvi Buon 1991!

Commodore Computer Club

COMMODORE COMPUTER CLUB

Direttore: Alessandro de Simone

Redazione / Collaboratori:
Davide Ardizzone - Claudio Baiocchi
Luigi Callegari - Umbero Colapicchioni
Donato De Luca - Carlo D'Ippolito
Valerio Ferri - Michele Maggi
Giancarlo Mariani - Domenico Pavone
Armando Sforzi - Dario Pistella
Fabio Sorgato- Valentino Spataro
Franco Rodella- Stefano Simonelli
Luca Viola

Direzione:
Via Mosè, 22 cap. 20090 OPERA (Mi)

Telefono 02 / 55.50.03.10
Fax 02 / 57.60.30.39
BBS 02 / 57.60.52.11

Pubblicità:
Leandro Nencioni (dir. vendite)
Via Mose', 22 20090 Opera (Mi)
tel. 02 / 55.50.03.10

Emilia Romagna:
Spazio E
P.zza Roosvelt, 4 cap. 40123 Bologna
Tel. 051 / 23.69.79

Toscana, Marche, Umbria
Mercurio s.r.l. Via Rodari, 9
S. G.nni Valdarno (Ar)
Tel. 055 / 94.74.44

Lazio, Campania
Spazio Nuovo
Via P. Foscari, 70
cap. 00139 Roma
tel. 06 / 81.09.679

Abbonamenti: Liliana Spina
Arretrati e s/w: Lucia Dominoni

Tariffe: Prezzo per copia L. 6000
Abbonamento annuo (11 fascicoli) L. 60000
Estero: L. 100000 - Indirizzare versamenti a:
Systems Editoriale Srl c/c 37952207 oppure
inviare comune assegno bancario non
trasferibile e barrato due volte a:
Systems Editoriale Srl (servizio arretrati)
Via Mose', 22
cap. 20090 OPERA (Mi)

Composizione: Systems Editoriale
La Litografica Srl Busto Arsizio (Va)

Registrazioni: Tribunale di Milano
n. 370 del 2/10/82

Direttore Responsabile: Michele Di Pisa

Spedizioni in abbonamento postale gruppo
III. Pubblicita' inferiore al 70%

Distributore: Parrini - Milano

Periodici Systems: Banca Oggi -
Commodore Club (disco) - Commodore
Computer Club - Commodore Computer
Club (disco, produzione tedesca) - Computer
- Computer disco - Electronic Mass Media
Age - Hospital Management - Jonathan -
Nursing '90 - PC Programm (disco) -
Personal Computer - Software Club
(cassetta ed. italiana) - Videoteca
VR Videoregistrare

Editoriale

Chi troppo vuole

Leggo spesso la rivista **Computer Products** (della "**Asian sources**") che, in effetti, non può essere definita propriamente una rivista.

Rappresenta, infatti, un immenso catalogo (si avvicina sempre più alle 1000 pagine), con cadenza mensile, sulle cui pagine si avvicendano inserzioni pubblicitarie di **case costruttrici** asiatiche (prevalentemente Taiwan, Korea, Hong Kong, Singapore e così cantando) che offrono i loro prodotti non certo all'utente finale, nè ai negozianti al dettaglio, ma addiritura agli importatori; oppure a chi può vantare fatturati a questi ultimi paragonabili. Il cliente "medio" dell'inserzione pubblicitaria di Computer Products è, insomma, chi acquista **almeno** un paio di centinaia di articoli alla volta (computers, schede varie e così via).

Che cosa offrono gli inserzionisti di Computer Products? Facile da immaginare: computers, schede di ogni tipo, tastiere, dischetti, monitors, interfacce, memorie, joy, mouse, connettori e perfino apparecchiature automatizzate per la loro produzione.

Alla stessa "serie" Asian Products appartengono altre riviste (tra cui Electronics) tutte accomunate dal desiderio di vendere, a chiunque lo voglia, prodotti orientali di ogni tipo: hi-fi, pelletteria, antifurti, cerniere lampo, giubbotti, lampade, telefoni, giocattoli, articoli di bellezza e tutto ciò, insomma, che troviamo in abbondanza nei nostri negozi.

In fondo a ciascun fascicolo vengono riportate, su ogni numero, le festività nazionali dei vari Paesi orientali (per evitare che un cliente capiti per affari in un giorno di chiusura totale), le modalità da seguire per prenotare un posto in aereo ed i numeri di telefono dei principali alberghi delle più importanti città; oltre, ovviamente, alle date delle numerose Fiere intercontinentali, sempre più affollate di acquirenti dai portafogli ben forniti.

Perfino il più sprovveduto ed occasionale lettore di Computer Products riesce ad immaginare che un tale sforzo organizzativo e pubblicitario non può essere un bluff (la pubblicazione festeggerà tra breve i 10 anni di crescente successo) e merita, quindi, di essere osservato sia pensando al presente sia, soprattutto, al futuro.

Naturalmente non ci riferiamo al successo della rivista, che è solo portavoce di una realtà produttiva estremamente aggressiva, ma a quello dei produttori orientali che hanno deciso di invadere il mercato planetario a dispetto di qualsiasi strategia protezionistica, o commerciale in senso lato.

Chi oggi, ad esempio, decide di produrre computers in occidente, potrà farlo tenendo conto che è assolutamente necessario offrire qualcosa di meglio di quanto un qualsiasi sconosciuto, ma avveduto (e soprattutto concorrenziale), produttore orientale è in grado di offrire.

Allargare le braccia, sconsolati, e dichiarare che non era possibile prevedere una concorrenza così spietata, un andamento del mercato diverso dalle aspettative e, comunque, una ineluttabilità degli eventi è, a nostro parere, falso o, quantomeno, frutto di assenza di intuizione.

Ma se una certa sovrabbondanza di manodopera nel campo informatico occidentale (ed italiano in particolare) poteva essere largamente prevista, discussa e fronteggiata con ben altre politiche competitive (avendo cura di evitare, magari, imbarazzanti offerte autunnali di computer quasi obsoleti con il 10% di sconto...) trascurare ancora il fenomeno asiatico può rappresentare, alla luce di quanto detto, quantomeno suicida.

In uno degli ultimi fascicoli, ad esempio, veniva posta, con il dovuto risalto, l'informazione che i colossi dei videogames del nostro pianeta (Sega e Nintendo, tanto per dire due nomi a caso) hanno deciso di offrire prodotti adeguati alle aspettative del mercato. E' probabile che, dato che ci sono, escogiteranno qualcosa per evitare il fenomeno delle pirateria in modo da incoraggiare le varie software house (che di questo male periscono) a sviluppare videogames più redditizi preferendo le nuove macchine a sistemi che, anche tecnologicamente parlando, non sono ormai più nè carne nè pesce.

Ostinarsi, inoltre, ad offrire ai rivenditori al dettaglio condizioni non propriamente vantaggiose, non fa altro che spingerli a considerare altre politiche commerciali di maggiore interesse, sia per se stessi sia, vivaddio, per l'utente finale. Molti rivenditori lo stanno già facendo, con notevole successo.

Tutti gli altri li seguiranno tra breve.

Alessandro de Simone

LA VOSTRA POSTA

(a cura di A. de Simone)

Telematica con il C/64

Alcuni lettori (tra cui **Bruno Di Gese** e **Roberto Politi**) chiedono se sia consigliabile collegarsi con il C/64 alle varie banche dati. La risposta, più volte suggerita su queste pagine, non può, ovviamente, sconsigliare collegamenti del genere. Alcuni riescono, anche a 1200 baud, ad effettuare collegamenti. I più, invece, sono costretti, per vari motivi, a limitarsi ai **300 baud**, con incremento notevole del costo delle bollette del telefono.

Pur se il desiderio di risparmiare sull'hardware è lodevole, non dobbiamo dimenticare che ciò che risparmiamo, oggi, sull'acquisto di un computer più idoneo a collegamenti telematici (leggi **Ms-Dos** ed **Amiga**) lo pagheremo, domani (ed ogni bimestre...) sulla bolletta SIP.

Indici di C.C.C.

Mi piacerebbe veder pubblicato un indice di tutti gli articoli pubblicati sulla vostra rivista.
(Marco Alzetta - Pordenone)

Tempo fa avevo iniziato l'improbo lavoro, ma non l'ho portato a termine per mancanza di tempo. La richiesta di Marco, tuttavia, mi ha fatto accendere una lampadina per stimolare i nostri lettori in un'ennesima sfida:

Utilizzate un Data Base qualunque (purchè "sotto" sistemi Ms-Dos o Amiga) per impostare un efficace archivio in cui l'utente possa rintracciare, con la massima facilità, un qualsiasi articolo o programma dopo aver indicato l'argomento richiesto.

Naturalmente possono partecipare alla "sfida" non solo lettori che posseggono un Amiga o un Ms-Dos compatibile, ma anche coloro che hanno collezionato tutti (o quasi) i numeri della nostra testata. Il file che ne risulterà (scritto secondo uno standard da concordare **preventivamente** per telefono) verrà, da noi, costantemente aggiornato mese per mese ed inserito nella nostra **BBS**, a disposizione di tutti.

Per informazioni sulla scelta dei campi del Data Base, telefonate in Redazione (**02/555.00.310**) al pomeriggio del giovedì.

Investire in competenze

Siamo un gruppo di ragazzi di Assisi, appassionati di informatica, e ci piacerebbe sviluppare software da vendere direttamente o tramite software house. Su quali sistemi conviene investire il nostro tempo (e denaro...)?
(Federico L. - Assisi)

Bella domanda, davvero. Purtroppo non indicate preferenze di alcun tipo e sono costretto ad andare a braccia.

Dunque: il motivo per cui qualcuno si debba sentire in dovere di offrirvi del denaro, è da ricercare nel vantaggio economico che può trarre dal vostro operato.

In parole molto più semplici, qualcuno vi paga se (e solo se) ci guadagna qualcosa.

Ad esempio, quando una s/w house distribuisce videogiochi, dal prezzo di vendita al pubblico (apparentemente esagerato) si devono detrarre: guadagno del dettagliante (25% circa, e non dite di no); trasporto e distribuzione; duplicazione dischetti; confezionamento; compenso per il disegnatore della confezione; guadagno della s/w stessa (a meno che non siano frati benedettini diditi alla beneficenza); giacenza magazzino per copie invendute; fatture dell'inevitabile campagna pubblicitaria; e (per ultime, badate bene!) competenze per i programmatori del gioco.

Se non è ancora chiaro, il compenso di un videogame inizia a divenire interessante se (e solo se) è possibile vendere, a circa 40 mila lire ciascuna, almeno 5 mila copie.

Se la s/w house non è sicura di vendere un numero sufficiente di copie, la conseguenza è semplice: rinuncia alla realizzazione del gioco, oppure diminuisce il compenso dei programmatori (gli altri anelli della catena non rinunciano facilmente alle loro percentuali).

Nel campo dei videogames, quindi, la vita è molto, molto dura; per non parlare, poi, della piaga della pirateria, che costringe ad investire massicciamente in campagne pubblicitarie e distribuzioni rapidissime (vedi U.S.A.) per recuperare denaro almeno da coloro che sono disposti ad acquistare **subito** il nuovo gioco piuttosto che **aspettare** per procurarselo copiato.

Videogames, oggi, vuol dire Amiga; Ms-Dos un po' meno.

E domani, quali macchine per videogames verranno distribuite? E che fine può fare la competenza accumulata su queste macchine? Meglio non pensarci.

Ne consegue che investire nel sistema **Ms-Dos** (in cui aziende prestigiose s/w ed h/w hanno speso, e continueranno a spendere, fiumi di denaro) può rappresentare, al giorno d'oggi, una sicurezza in più e, soprattutto, la possibilità di entrare in un mercato enormemente più vasto ed in continua crescita; basta pensare allo sviluppo delle **reti**, tanto per uscire dal ristretto ambito dei sistemi **stand alone**.

Stampanti

Posseggo un C/64 ed un drive 1541; vorrei acquistare una stampante che sia compatibile con i programmi grafici del mio computer (Geos, Speed Script, Speed Calc, ecc.). Che modello mi consigliate?
(Vittorio Dominici - S.to F.no)

Il C/64 ha un modo particolare di gestire la grafica in alta risoluzione, che le stampanti da collegare al computer devono rispettare rigorosamente. Le stampanti Commodore, ovviamente, sono compatibili con il C/64; tuttavia, stampanti di altre marche (dichiarate, dal fabbricante, **Mps-803 compatibili**) possono egualmente risolvere il problema.

E' bene sottolineare, però, che **quasi sempre** le stampanti (**non** compatibili 803) collegate al C/64 mediante varie interfacce, presentano problemi gravi di incongruenza, soprattutto nel caso di stampa di schermate grafiche in alta risoluzione.

Per risolvere il problema del futuro utilizzo con altri computer (Ms-Dos oppure Amiga) ti consiglio, quindi, di procurarti una stampante che presenti entrambe le interfacce: seriale C/64 e Centronics, di marca, inutile dirlo, Commodore.

Operare su sistemi Ms-Dos significa, però, avere a che fare, prevalentemente, con applicazioni professionali, in cui la grafica ed il sonoro possono venire mortificati in favore di esigenze di ben altra portata.

In conclusione? Divertirsi a programmare marzianetti da colpire, oppure escogitare un sistema per inviare, in tempo reale, un documento in una rete di Personal Computer? Ai posteri l'ardua sentenza...

Odio e Amore

Odio l'Amiga e ritengo che piaccia solo ai Redattori delle riviste di informatica che possono disporre di programmi originali (e gratis) e se ne infischiano dei poveracci che non possono spendere soldi per programmare decentemente. Viva l'immortale C/64!
(A. Blasio & C. Hansen)

Non sono d'accordo sui toni delle due lettere (qui estremamente condensate); devo riconoscere, ovviamente, che i Redattori usufruiscono di particolari agevolazioni, ma non sono d'accordo sul fatto che tutto viene offerto gratuitamente. Finora ho speso diversi **milioni** per acquistare vari computer, monitor, stampanti (ed altro) che mi consentissero di affrontare i nuovi argomenti imposti dai lettori.

Ultimamente ho acquistato un 386 "da parata" (e tra breve comprerò, sempre con il mio denaro, una stampante laser Postscript) proprio per aumentare la produttività del lavoro e soddisfare le richieste dei lettori, che prevedo numerose. Fra quache anno tutte le apparecchiature, che ora ho acquistato, varranno molto meno della metà di quanto oggi spendo, pur se le ho ottenute con un sensibile sconto.

Mi preme, però, insistere sul fatto che per il C/64, ormai, è stato detto tutto; noi stessi incominciamo, te ne sarai accorto, a **ripeterci**. Pertanto (e parlo contro il mio interesse, e della Systems Editoriale) non ha senso acquistare la nostra rivista se ci si vuole ostinare ad usare solo il C/64: tutto ciò che è stato finora pubblicato rappresenta una vera e propria enciclopedia, che ha esplorato l'angolo più recondito del piccolo computer Commodore (alta risoluzione, aggiunta di comandi Basic, sprite sullo schermo, musiche nell'interrupt, raster, implementazione di istruzioni bizzarre, e così via).

Davvero volete continuare a spendere denaro per procurarvi riviste che, in fin dei conti, dicono ormai sempre e solo le stesse cose? Contenti voi...

Ms-Dos incompatibile

Con il mio (vecchio) computer XT Ms-Dos spesso non riesco a leggere i dischetti provenienti dal nuovo computer At Ms-Dos che un mio amico ha acquistato di recente; il contrario, invece, non capita mai. C'è forse qualche incompatibilità tra le varie versioni del Dos?
(Alfredo B. Milano)

Assolutamente no, ci mancherebbe altro! La compatibilità di un sistema serio (come, appunto, l'Ms-Dos) è totale e garantita (guai se non lo fosse). C'è da notare, però, che la compatibilità delle macchine è garantita **verso il basso** e non **verso l'alto**; vediamo di spiegarci.

In un qualsiasi momento dello sviluppo informatico non è possibile sapere, ovviamente, che cosa ci riserva il futuro. Ai tempi dei primi esemplari Ms-Dos, la tecnologia offriva la possibilità di memorizzare dati e programmi su dischetti da 360 Kbyte; gli sviluppatori non potevano **prevedere** (potevano solo **intuirlo**) che il progresso avrebbe consentito la fabbricazione di testine e supporti magnetici in grado di trattare **1.2 megabyte** per dischetto.

Quando vennero fabbricati i drive da 1.2 mega, il nuovo software di gestione della periferica ebbe, quindi, non solo il compito di gestire in modo diverso la maggiore quantità di dati, ma anche (e soprattutto) quello di continuare a "riconoscere" (e "trattare" di conseguenza) TUTTI i dischetti prodotti dalle precedenti versioni del Dos, compresi quelli, antichissimi, che consentivano la gestioni dei dati su una sola faccia del dischetto(!).

I moderni computer, in conclusione, sono in grado di trattare i dati provenienti da una **qualsiasi** versione dell'Ms-Dos, pur se risalente agli inizi degli anni '80: questo particolare giustifica, quindi, il successo del *sistema* Ms-Dos rispetto a *sistemucoli* di altro calibro; non è necessario buttare via tutto il software (ed i file con esso prodotti) accumulato, quando il progresso tecnologico impone cambiamenti dell'hardware.

La compatibilità, però, è garantita verso il basso (cioè con versioni precedenti del Dos) e non verso l'alto (con le future versioni del Dos) proprio perchè, lo ripeto, non possiamo sapere che cosa ci riserva il futuro.

E veniamo al problema, escludendo il caso, oltremodo banale, dell'impossibilità, da parte di un drive da 360 K, di leggere dischetti da 1.2 megabyte....

Sembrerebbe, quindi, che l'impossibilità del tuo (vecchio) computer di leggere dischetti provenienti da un (nuovo) computer AT sia dovuto, in parte, all'incompatibilità verso l'alto cui abbiamo accennato: lo dimostra il fatto che il tuo amico legge TUTTI i dischetti provenienti dal tuo computer.

Perchè, allora, il tuo XT legge alcuni dischi sì ed altri no? La incompatibilità dovrebbe esserci sempre, oppure mai.

Il problema risiede nella differente dimensione delle testine dei due tipi di drives (360 Kbyte ed 1.2 megabyte). Quest'ultimo tipo, inevitabilmente, ha una testina di lettura / scrittura più piccola; i "solchi" che una testina di questo tipo traccia su un dischetto, quindi, sono di dimensioni minori rispetto a quelli tracciati da un vecchio drive da 360 K (che ha una testina più larga).

Formattiamo, con il computer AT, un dischetto **vergine** con la modalità da 360 K; in seguito scriviamo file, cancelliamoli, sovrascriviamo e, insomma, "usiamolo" come più ci aggrada, ma sempre con il computer AT. Il dischetto può

SYSTEMS EDITORIALE PER TE

La voce
Aggiunge al C/64 nuovi comandi Basic che consentono sia di far parlare il computer, sia di farlo Cantare! Diversi esempi allegati.

Cassetta: L. 12000 - Disco: L. 15000

Raffaello
Un programma completo per disegnare, a colori, con il C/64: linee, cerchi, quadrati, eccetera. Valido sia per disegno a mano libera che geometrico.

Cassetta: L. 10000

Oroscopo
Devi solo digitare la data di nascita e le coordinate geografiche del luogo che ti ha dato i natali. Vengono quindi elaborate le varie informazioni (case, influenze dei segni astrali, eccetera) e visualizzato un profilo del tuo carattere. Valido per qualsiasi anno, è indicato sia agli esperti sia ai meno introdotti. E' allegata una tabella delle coordinate delle più note città italiane e l'elenco delle ore legali in Italia dal 1916 al 1978.

Cassetta: L. 12000 - Disco: L. 12000

Computer Music
Cassetta contenente numerosi brani di successo da far eseguire, in interrupt, al tuo C/64 sfruttando, fino in fondo, il suo generatore sonoro (SID).

Cassetta: L. 12000

Gestione Familiare
Il più noto ed economico programma per controllare le spese e i guadagni di una famiglia.

Cassetta: L. 10000 - Disco: L. 10000

Banca Dati
Il più noto ed economico programma per gestire dati di qualsiasi natura.

Cassetta: L. 10000 - Disco: L. 10000

Matematica finanziaria
Un programma completo per la soluzione dei più frequenti problemi del settore.

Cassetta: L. 10000 - Disco: L. 20000

Analisi di bilancio
Uno strumento efficace per determinare con precisione i calcoli necessari ad un corretto bilancio.

Cassetta: L. 10000 - Disco: L. 20000

Corso di Basic
Confezione contenente quattro cassette per imparare velocemente le caratteristiche delle istruzioni Basic del C/64 e i rudimenti di programmazione. Interattivo.

Cassetta: L. 19000

Corso di Assembler
Un corso completo su cassetta per chi ha deciso di abbandonare il Basic del C/64 per addentrarsi nello studio delle potenzialità del microprocessore 6502. Interattivo.

Cassetta: L. 10000

Logo Systems
Il linguaggio più facile ed intuitivo esistente nel campo dell'informatica; ideale per far avvicinare i bambini al calcolatore.
Diversi esempi allegati.

Cassetta: L. 6500

Compilatore Grafico Matematico
Uno straordinario programma compilatore, di uso semplicissimo, che permette di tracciare, sul C/64, grafici matematici Hi-Res ad altissima velocità. Esempi d'uso allegati.

Cassetta: L. 8000

Emulatore Ms-Dos e Gw-Basic
Un prodotto, unico nel suo genere, che permette di usare, sul C/64 dotato di drive, la sintassi tipica del più diffuso sistema operativo del mondo. Ideale per studenti.

Solo su disco: L. 20000

Emulatore Turbo Pascal 64
Permette di usare le più importanti forme sintattiche del linguaggio Turbo Pascal (anche grafiche!) usando un semplice C/64 dotato di drive.
Ideale per studenti.

Disco: L. 19000

Speciale drive
Questo speciale fascicolo costituisce una guida di riferimento per le unità a disco del C64/128.
Comprende anche un velocissimo turbo-disk più la mappa completa della memoria del drive.

Fascicolo + disco: L. 12000

Utility 1
Un dischetto pieno zeppo di programmi speciali per chi opera frequentemente con il drive.

Disco: L. 12000

Utility 2
Seconda raccolta di utility indispensabili per realizzare sofisticate procedure di programmazione.

Disco: L. 15000

Graphic Expander 128
Per usare il C/128 (in modo 128 e su 80 colonne) in modo grafico Hi-res. Aggiunge nuove, potenti istruzioni Basic per disegnare in Hi-Res con la massima velocità in modalità 80 colonne.

Disco: L. 27000

Directory
Come è noto, a partire dal N. 10 di "Software Club" (la rivista su disco per l'utente dei "piccoli" computer Commodore), vengono riportati tutti i listati, in formato C/64-C/128, pubblicati su "Commodore Computer Club".
In precedenza tali listati venivano inseriti, mensilmente, in un dischetto, di nome "Directory", che oltre ai programmi di C.C.C. ospitava decine di altri file tra cui musiche nell'interrupt, giochi, listati inviati dai lettori e altro.
Ogni disco, dal prezzo irrisorio, contiene quindi una vera miniera di software. Ordinando i dischetti di "Directory" si tenga conto che al N. 1 corrispondeva il contenuto del N. 34 di "Commodore Computer Club", al N. 2 il N. 35 e così via.

Ogni dischetto: L. 10000

Super Tot '64
La nuova e completa edizione del programma Tot 13 con tutti i sistemi di riduzione e di condizionamento.
Ampia sezione dedicata alla teoria.

fascicolo + disco: L. 15000

Amiga Totospeed
Finalmente anche per Amiga un programma orientato alla compilazione delle schedine totocalcio.
Fai tredici conil tuo Amiga.

disco: L. 20000

SYSTEMS EDITORIALE PER TE

Disk'o'teca
Grazie a questa nutrita raccolta di brani musicali potrete divertirvi ascoltando i migliori brani prodotti dai vostri beniamini, oltre a una serie di composizioni prodotte "in casa".
In omaggio un bellissimo poster di Sting.
Disco: L.15.000

Assaggio di primavera
Esclusivo!
In un'unica confezione potrete trovare ben due cassette di videogiochi assieme a un comodo e funzionale joystick.
Cassette: L. 15.000

LIBRI TASCABILI

64 programmi per il C/64
Raccolta di programmi (giochi e utilità) semplici da digitare e da usare. Ideale per i principianti. **(126 pag.)**
L. 4800

I miei amici C/16 e Plus/4
Il volumetto, di facile apprendimento, rappresenta un vero e proprio mini-corso di Basic per i due computer Commodore. Numerosi programmi, di immediata digitazione, completano la parte teorica. **(127 pag.)**
L. 7000

62 programmi per C/16, Plus/4
Raccolta di numerosi programmi, molto brevi e semplici da digitare, per conoscere più a fondo il proprio elaboratore.
Ideale per i principianti. **(127 pag.)**
L. 6500

Micro Pascal 64
Descrizione accurata della sintassi usata dal linguaggio Pascal "classico". Completa il volume un programma di emulazione del PL/O sia in formato Microsoft sia in versione C/64 (da chiedere, a parte, su disco). **(125 pag.)**
L. 7000

Dal registratore al Drive
Esame accurato delle istruzioni relative alle due più popolari periferiche del C/64. Diversi programmi applicativi ed esempi d'uso. (94 pag.)
L. 7000

Il linguaggio Pascal
Esame approfondito della sintassi usata nel famoso compilatore. (112 pag.)
L. 5000

Simulazioni e test per la didattica
Raccolta di numerosi programmi che approfondiscono e tendono a completare la trattazione già affrontata sul precedente volume. **(127 pag.)**
L. 7000

Dizionario dell'Informatica
Dizionario inglese-italiano di tutti i termini usati nell'informatica. (Edizione completa). **(385 pag.)**
L. 10000

Word processing: istruzioni per l'uso
Raccolta delle principali istruzioni dei più diffusi programmi di w/p per i sistemi Ms-Dos: Word-Star, Samna, Multimate Advantage, Word 3. **(79 pag.)**
L. 5000

Unix
Un volumetto per saperne di più sul sistema operativo professionale per eccellenza.
Un necessario compendio per l'utente sia avanzato che inesperto **(91 pag.)**
L. 5000

ABBONAMENTO

Commodore Computer Club
11 fascicoli: L. 60.000

ARRETRATI

Ciascun numero arretrato di C.C.C. L. 6.000

Come richiedere i prodotti Systems

Coloro che desiderano procurarsi i prodotti della Systems Editoriale devono inviare, oltre alla cifra risultante dalla somma dei singoli prodotti, L. 3500 per spese di imballo e spedizione, oppure L. 6000 se si desidera la spedizione per mezzo raccomandata.
Le spese di imballo e spedizione sono a carico della Systems se ciascun ordine è pari ad almeno L. 50000.
Per gli ordini, compilare un normale modulo di C/C postale indirizzato a:

C/C Postale N. 37 95 22 07
Systems Editoriale Srl
Via Mosè, 22
20090 Opera (MI)

Non dimenticate di indicare chiaramente, sul retro del modulo (nello spazio indicato con "Causale del versamento"), non solo il vostro nominativo completo di recapito telefonico, ma anche i prodotti desiderati ed il tipo di spedizione da effettuare.
Per sveltire la procedura di spedizione sarebbe opportuno inviare, a parte, una lettera riassuntiva dell'ordine effettuato, allegando una fotocopia della ricevuta del versamento.
Chi volesse ricevere più celermente la confezione deve inviare la somma richiesta mediante assegno circolare, oppure normale assegno bancario (non trasferibile o barrato due volte) intestato a:

Systems Editoriale
Milano

ora esser letto senza problemi dal vecchio drive XT.

Se, però, un dischetto è stato creato (oppure "usato", nel senso che ha subito una o più **scritture**) da un vecchio drive da 360K, i solchi da questo creato risultano più larghi di quelli precedenti. Nessun problema, però, nel ri-trattare il disco con un computer AT. Se, però, ora cancelliamo (con il drive dell'AT) un file che era stato scritto con il vecchio drive, la testina dell'AT non modificherà per intero il vecchio solco, a causa della ridotta dimensione della testina. Ne consegue che se per un AT non vi sono problemi di lettura e scrittura, la testina di un vecchio drive è costretta a leggere un solco nella cui parte centrale è presente un altro solco, e segnala errore. A dire la verità, i nuovi drive da 360 Kbyte non presentano questo problema. Per i vecchi computer esiste, però, un programma di pubblico dominio (dal nome **BlukErase**) che consente ad un computer AT di formattare i dischetti non vergini (cioè già usati) in un modo tale da evitare errori di lettura da parte di vecchi drive. A patto, ovviamente, di non trattarli con un vecchio drive e, in seguito, trattarli ancora con un AT.

Etichette autoadesive

Usando i miei dischetti mi capita spesso di apportare modifiche, cancellazioni e variazioni in genere che vorrei riportare anche sulle etichette autoadesive sovrapposte ai floppy. Purtroppo non riesco a trovare in vendita etichette che presentino la caratteristica di staccarsi con facilità; mi trovo ad operare, così, con floppy dall'aspetto... sporco a causa di brandelli di vecchie etichette che non riesco a rimuovere. Dove si possono acquistare etichette per floppy disk?
(Anonimo disk jockey)

Anche i miei dischetti, purtroppo, sembrano spesso prelevati dal cestino della carta straccia. I negozi (molto) specializzati di forniture per uffici dovrebbero avere il tipo di etichette che richiedi, ma non so a che prezzo.

Personalmente ho risolto il problema in modo forse poco elegante, ma estremamente efficace (e, soprattutto, economico). Chiedi, in cartoleria, un rotolo di nastro adesivo del tipo utilizzato dai disegnatori per fissare i fogli al tavolo da disegno; i nastri di questo tipo possiedono due preziose caratteristiche: anzitutto si distaccano con la massima facilità (e SENZA lasciar traccia) sia dalle superfici dei floppy da 5.25 sia da quelle del formato 3.5.

Inoltre sono di carta e consentono, quindi, la scrittura di messaggi. Servendosi di un comune taglierino possono essere tagliati secondo la lunghezza che desideri. Le altezze del nastro sono di misura variabile. La più piccola è di mm. 18, più che sufficiente per scrivere almeno il nome dei files e delle directory più importanti contenute nel dischetto.

Non è il massimo dell'eleganza, ma chi si contenta gode!

Musica & Grafica, Ms-Dos o Amiga?

Mi piacerebbe molto sostituire il mio vecchio C/64 con un computer più moderno, senza rinunciare all'aspetto ludico che mi ha sempre appassionato. Il dubbio è il solito: è meglio un Ms-Dos compatibile forse più costoso, ma di potenzialità professionali (e di vita media sicuramente più lunga) oppure un più economico Amiga dotato di grafica e musica insuperabili?
(Muzio A. - Torre G.)

Se la grafica e la musica ti interessano molto, forse non sai che, ormai, il sistema Ms-Dos non è più secondo a nessuno (e quando dico nessuno, intendo proprio nessuno).

Cominciamo dalla grafica. Oggi una scheda **Super VGA** offre (a prezzo, in verità, un tantino alto per un semplice hobbysta) una risoluzione di 1024 x 768 ed una scelta di 256 colori. Amiga offre 4096 colori, d'accordo, ma hai mai visto le immagini riprodotte da una "semplice" VGA (800 x 600)? Se ti càpita, recati presso un qualsiasi negozio specializzato per osservare uno dei tanti **Demo** che mettono in luce le caratteristiche delle schede grafiche: sembrano vere e proprie diapositive a colori.

Ma ciò che può sconfiggere l'Amiga su tutti i fronti è rappresentato, paradossalmente, dalle capacità sonore offerte dalle moderne schede musicali / vocali per sistemi Ms-Dos.

Mi riferisco, in particolare, alla ben nota scheda **Sound Blaster** che ormai viene venduta a prezzo sufficientemente contenuto da un qualsiasi rivenditore specializzato. A proposito: è doveroso segnalare la Circe Electronics di Milano (tel. 02/64.27.410 oppure 02/26.11.20.24) che ci ha consentito di tenere in prova un esemplare del Sound Blaster, che offre in vendita al prezzo più basso che ci è capitato di rintracciare.

La scheda di cui parliamo può essere montata su un **qualsiasi Ms-Dos compatibile** (XT, AT 286 / 386sx / 386dx) anche se i risultati migliori si ottengono con i computer dall'80286 in su.

Contiene, al suo interno:

* 11 voci di Fm music;
* due modi di funzionamento (9 sounds oppure 6 melodie e 5 ritmi);
* un canale di digitalizzazione vocale (con effetti speciali);
* possibilità di intervento della CPU oppure modo DMA (utilizzo diretto);
* amplificatore **stereo** da 4 watts per canale (è presente, sul retro, sia il jack stereo per collegamento alla cuffia o ad altra fonte di amplificazione, sia un regolatore di volume);
* ingresso microfonico (per effettuare digitalizzazioni vocali mediante un comune **microfono** da 600 ohm di impedenza);
* porta **joy** analogica;
* interfaccia **MIDI** (richiede solo un connettore esterno);
* zoccolo per il collocamento di altri due chips utili per la generazione di altra musica sintetizzata.

Nella confezione di Sound Blaster è presente software per programmi dimostrativi (e di intrattenimento) che sono realmente fantastici e non fanno rimpiangere nulla delle capacità sonore di Amiga. La scheda è **AdLib Music Card** compatibile, vale a dire che è in grado di riconoscere i programmi scritti per utilizzare l'altra nota scheda musicale (di questo tipo, infatti, ve ne sono diverse). Moltissimi **videogames** prodotti dalle più note s/w house sono in grado di "riconoscere" la presenza della scheda sul computer e di sfruttarla adeguatamente riproducendo **voci sintetizzate**, **effetti sonori**, **musiche stereofoniche** e così via.

Le potenzialità della scheda Sound Blaster ci sono sembrate tanto convincenti (ed il suo **prezzo al pubblico** talmente invitante) che abbiamo preso la decisione di "aprire", tra breve, una vera e propria **rubrica dedicata alle schede musicali Ms-Dos** ed alla loro programmazione.

Ingegneria e C/128

Posseggo il C/128, che non intendo assolutamente sostituire, e sono alla ricerca di un programma in grado di elaborare (e stampare) telai, verifica antisismica, setti in cemento armato, isolamento termico, impianti elettrici civili.

(Ing. Paolo P.)

Nient'altro? Bè, in questo caso la risposta è un semplice **no**, altrimenti sarebbe stata **assolutamente no**.

Scherzi a parte, non dubito che, negli anni precedenti, qualcuno abbia sviluppato e distribuito programmi di ingegneria per C/64 (e, *forse*, per C/128), soprattutto per l'elaborazione di dati piuttosto semplici.

Ma progettare un telaio con verifica antisismica (e corrispondente stampa dei "ferri") non è uno scherzo, caro collega (sono ingegnere anch'io e conosco le difficoltà di realizzare un programma di tale portata).

Chi, quindi, oggi si pone all'opera per la stesura di un programma complesso, non lo fa certamente su un C/128 (nè, consentitemi, su un Amiga) ma solo su un **Ms-Dos** compatibile. Il motivo è semplicissimo. Alcuni calcoli richiedono elaborazioni complesse che possono essere eseguite, in un tempo accettabile (il tempo è denaro!), solo avendo a disposizione un computer **veloce** (magari dotato di processore matematico), capace di gestire dati in **doppia precisone** (se parliamo di verifica antisismica, le approssimazioni potrebbero portare a risultati imbarazzanti in caso di terremoto...), in grado di "trattare" una miriade di dati (senza commettere errori di alcun tipo) e, infine, dotato di interfaccia software per stampare i risultati grafici su carta (dimensioni dei ferri e loro piegatura, loro allineamento, posizionamento corretto di staffe, dimensionamento di pilastri, travi, solai e così via).

Il secondo motivo per cui un programmatore si pone all'opera su un **vero** computer (e non su un giocattolo) è rappresentato dal fatto che il software prodotto verrà venduto presumibilmente(!) ad uno studio **professionale**, sicuramente attrezzato con computer Ms-Dos compatibili.

Sì e NOT

Non riesco a capire in che modo agisca l'istruzione NOT. Questa dovrebbe convertire i bit di un byte da 0 a 1 (e viceversa); il numero decimale 10 (che in binario è 0000 1010) dovrebbe diventare, con l'istruzione* Not 10, *il numero 245 (cioè: 1111 0101). Con il comando* Print Not (10), *invece, ottengo l'incomprensibile valore -11. Come mai?

(Maurizio - Viterbo)

Digita il seguente listatino:

```
100 rem esame dell'istruzione not
110 input "valore (0/127)"; a
115 if a > 127 then print "Errore": end
120 x=a: gosub 150: print, " numero:...", x
130 x=not a: gosub 150
140 print, " e suo not: ", x: end
150 for i=7 to 0 step -1
160 y=2^(i) and x
170 if y > 0 then print "1";: goto 190
180 print "0";
190 next: return
```

Digitando **10** (in risposta all'Input di riga 110), si ottiene...

0000 1010 numero: 10
1111 0101 e suo Not: -11

Digitando, ora, **-11**, otterrai...

1111 0101 numero: -11
0000 1010 e suo Not: 10

Infine, con **245** (eliminando la riga 115) si visualizza...

1111 0101 numero: 245
0000 1010 e suo Not: -246

Dunque, al numero binario 0000 1010 corrispondono sia il valore decimale **10** sia **-246**. Allo stesso modo, al numero binario 1111 0101 sembrano corrispondere sia **-11** che **245**.

Il fatto è che il comando **Not** non si limita a cambiare lo **0** in **1** (e viceversa), ma effettua il complemento a due del valore numerico cui viene applicato.

Il complemento a due di un qualsiasi numero intero viene determinato eseguendo il complemento al bit e sommando 1 al numero così ottenuto.

Il complemento a 2 del numero 10, al esempio, viene così determinato:

0000 1010 (dec. 10)
1111 0101 (complem.)
0000 0001 (somma +1)
1111 0110 (risultato -10)

Questo numero, nel campo dei numeri negativi, vale appunto **-10**.

Vediamo di chiarire meglio il concetto: se si desidera considerare solo valori positivi, con 8 bit è possibile gestire 256 valori (da 2^0 a 2^8). Se, però, rinunciamo ad una parte di questo intervallo, possiamo assegnare al primo bit (cioè, il settimo, se li numeriamo da 0 a 7) il significato di segno. In questo modo, ad esempio, il numero binario...

000 0001

...vale 1, mentre il valore decimale -1 viene rappresentato dal numero binario...

1111 11111

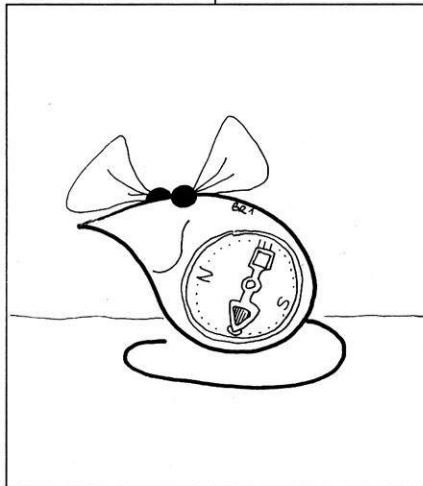
A questo numero binario si perviene effettuando il complemento a due del valore 1. Cioè:

0000 0001 (1 dec.)
1111 1110 (complem.)
0000 0001 (somma 1)
1111 1111 (risultato)

L'(apparente) valore decimale corrispondente (255, appunto), indica il valore -1 nel caso di una convenzione di numeri negativi e positivi.

Procedendo a "ritroso" si ottiene...

1111 1111 (= -1)
1111 1110 (= -2)
1111 1101 (= -3)
...
1111 0101 (= -11)
(...eccetera).

Quest'ultimo numero, guarda caso, è proprio quello che viene fuori effettuando **Not 10** e che viene visualizzato anche negando, bit per bit, il valore 10 trasformato, preventivamente, in binario. Per approfondire l'argomento, ti consiglio di studiare a fondo lo specifico argomento (complemento a due) su un qualsiasi libro di testo scolastico di informatica.

di Alessandro de Simone

RICOMINCIO DAL DOS

Chi inizia ad operare con un calcolatore, e con un Ms-Dos compatibile in particolare, non sempre trova a disposizione argomenti semplici e, perchè no?, divertenti. In queste pagine, quindi...

Dopo aver estratto il computer dal suo imballaggio, il principiante, anche se di buona volontà, si vede costretto, a causa della carenza di informazioni, a far girare i vari programmi a disposizione senza rendersi conto di come effettivamente funzionino.

Dal momento che non ci si accontenta di formattare dischetti e di realizzare output di pacchetti applicativi come un **W/P** o **Spreadsheet**, (per quanto utili siano) il principiante si reca subito nella più vicina libreria per scegliere un paio di volumi che gli consentano di conoscere in modo approfondito il computer appena acquistato.

A volte, però, la delusione è grande: si rischia, infatti, di acquistare un libro il cui contenuto altro non è se non le istruzioni già presenti sul libretto di istruzioni; oppure le sue pagine sono infarcite di frasi ostiche o di lunghi(ssimi) programmi scritti in linguaggi sconosciuti.

La tendenza a scoraggiarsi prende quindi il sopravvento e, credeteci, questa fase l'hanno attraversata tutti coloro che operano nel campo.

Queste pagine, pertanto, offriranno lo spunto per tirar fuori, dal nostro **Ms-Dos compatibile**, tutte le sue risorse, operando, soprattutto, mediante esempi pratici, di semplicissima attuazione, che si concluderanno sempre con pratiche applicazioni.

Naturalmente non pretendiamo di offrire una trattazione completa su ciascun argomento proposto; del resto, i suggerimenti che ci capiterà di divulgare, saranno un'ottima scusa per considerarli come l'inizio di vari esperimenti da compiere a tu per tu con il nostro elaboratore personale.

Perche' il DOS

Pochi ricordano che il nucleo centrale attorno a cui ruota qualsiasi applicazione di informatica è il **Sistema Operativo** (S. O.) che, a seconda dei casi, può essere residente su diversi supporti fisici. Accenneremo soltanto alle Rom (circuiti integrati) e alle memorie di massa, tipicamente i dischi.

Al sistema operativo per eccellenza è stato attribuito, non a caso, il nome **Ms-Dos**; delle cinque lettere, le prime due indicano la software house di provenienza (**Microsoft**), mentre le altre tre, acronimo di **D**isk **O**perative **S**ystem, rivelano, senza ombra di dubbio, che il sistema si regge interamente sulla corretta ed intensiva gestione dei dati su disco.

Uno studio, approfondito o superficiale che sia, non può prescindere da questa fondamentale considerazione anche perchè, prima o poi, ci si scontra sempre

```
                     Absolute sector 0000000, System BOOT

Displacement ----------------- Hex codes--------------------    ASCII value

0400(0190)   C3 0D 0A 44 69 73 63 6F 20 70 72 69 76 6F 20 64   +  Disco privo d
0416(01A0)   69 20 73 69 73 74 65 6D 61 0D 0A 50 72 65 6D 65   i sistema  Preme
0432(01B0)   72 65 20 75 6E 20 74 61 73 74 6F 0D 0A 00 0D 0A   re un tasto
0448(01C0)   44 69 66 65 74 74 6F 20 69 6E 20 63 61 72 69 63   Difetto in caric
0464(01D0)   61 6D 65 6E 74 6F 0D 0A 00 49 42 4D 42 49 4F 20   amento   IBMBIO
0480(01E0)   20 43 4F 4D 49 42 4D 44 4F 53 20 20 43 4F 4D 00    COMIBMDOS  COM
0496(01F0)   00 00 00 00 00 00 00 00 00 00 00 00 00 00 55 AA                 U
```

con protocolli di I/O, tracce, settori, cluster, file; insomma, il disco ci sarà sempre tra i piedi.

Tanto vale fare subito la sua conoscenza attraverso un paio di divertenti applicazioni.

Formattando

E' superfluo sottolineare che, in questa sede, daremo per scontate alcune conoscenze, dal momento che possono esser rintracciate facilmente sul libretto di istruzioni del vostro computer.

Ci limiteremo a ricordare, semmai, che ora andiamo sul pratico; sospendete quindi la lettura, procuratevi un dischetto-cavia da 3 pollici e mezzo o da 5 pollici 1/4 (per gli imminenti esperimenti sadomaso), accendete il computer e caricate il sistema in modo, insomma, che compaia il familiare Prompt. Fatto?

Bene, formattare un dischetto significa, in altre parole, prepararlo ad accettare dati che, in seguito, si presume che verranno inviati. La prima cosa che ci preme, quindi, è sapere che cosa succede, ad un dischetto, durante la sua formattazione e, soprattutto, quale sarà il suo "aspetto" dopo l'operazione.

Per rendercene conto fino in fondo sarà necessario seguire **alla lettera** tutti gli step che verranno indicati, senza ometterne alcuno; capirete dopo il perchè di tale pignoleria. Dunque, **formattate** il dischetto vergine nel formato più semplice possibile, senza riportarvi il sistema (cioè i file nascosti), ma assegnando il nome. In parole più semplici, attivate il drive A, inserite il disco sistema (oppure "raggiungete" la subdirectory dell'hard disk in cui è presente il comando **Format**) e digitate...

format a:/v

..in modo da ottenere, alla fine, un dischetto dotato di nome (label).

Coloro che posseggono un **AT compatibile**, dotato di drive da **1.2 mega**, dovranno formattare trattandolo come un drive da 360 K. Vale a dire:

format a:/4/v

Alla richiesta di inserire il floppy vergine nel **drive A**, sostituite il disco sistema con quello cavia e premete Return, stando ben attenti, durante la formattazione, ai messaggi che verranno visualizzati.

A seconda del Dos che possedete, infatti, compariranno messaggi in italiano o in inglese ma, tutti, vi faranno capire che, per ciascun **cilindro**, vengono for-

```
PC Tools Deluxe R4.22
------------------------------Sector Edit Service--------------------------------
Path=B:
                     Absolute sector 0000000, System BOOT

Displacement ----------------- Hex codes--------------------    ASCII value
 0256(0100)   39 7C A3 3F 7C BB 00 07 A1 39 7C E8 44 00 A1 18   9¦ ?¦=   9¦ D  ^
 0272(0110)   7C 2A 06 3D 7C 40 50 E8 54 00 58 72 CF 28 06 3E   ¦* =¦@P T Xr ( >
 0288(0120)   7C 76 11 01 06 39 7C 83 16 2B 7C 00 F7 26 0B 7C   ¦v<  9¦  +¦  & ¦
 0304(0130)   03 D8 EB D4 8A 2E 15 7C 8A 16 FD 7D 8B 1E 3F 7C        . ¦   } ^?¦
 0320(0140)   EA 00 00 70 00 AC 0A C0 74 23 B4 0E B3 FF CD 10      p   +t#+ ¦ =>
 0336(0150)   EB F3 8B 16 2B 7C F7 36 18 7C FE C2 88 16 3D 7C       +¦ 6^¦ +  =¦
 0352(0160)   33 D2 F7 36 1A 7C 88 16 2A 7C A3 3B 7C C3 B4 02   3  6 ¦  *¦ ;¦++
 0368(0170)   8B 16 3B 7C 8A EA D0 CE D0 CE 80 E6 C0 8A 0E 3D     ;¦          +  =
 0384(0180)   7C 80 E1 3F 0A CE 8A 36 2A 7C 8A 16 FD 7D CD 13   ¦  ?   6*¦   }=
 0400(0190)   C3 0D 0A 4D 61 20 73 65 69 20 6D 61 74 74 6F 3F   +  Ma sei matto?
 0416(01A0)   20 4D 65 74 74 69 20 69 6C 20 64 69 73 63 6F 20    Metti il disco
 0432(01B0)   64 69 20 73 69 73 74 65 6D 61 2E 0D 0A 00 0D 0A   di sistema.
 0448(01C0)   44 69 66 65 74 74 6F 20 69 6E 20 63 61 72 69 63   Difetto in caric
 0464(01D0)   61 6D 65 6E 74 6F 0D 0A 00 49 42 4D 42 49 4F 20   amento   IBMBIO
 0480(01E0)   20 43 4F 4D 49 42 4D 44 4F 53 20 20 43 4F 4D 00    COMIBMDOS  COM
 0496(01F0)   00 00 00 00 00 00 00 00 00 00 00 00 00 00 55 AA                 U
Sector updated - Press any key to continue
 ^ v -> <- = cursor  F1=swap entry area  F5=update  F6=cancel update  ESC=exit
Home=first pos  End=last pos  PgUp=1st half  PgDn=2nd half
```

mattati due **lati** del disco (numerati con **0** ed **1**). Vedremo perchè, in seguito, è interessante averlo notato. Il numero di cilindri risulterà 40, magari numerati da **0** a **39** (80 se operate con un drive da 3.5 pollici). Come nome assegnate: "**Leucippo**" (in modo da evitare il solito "Pippo", di cui sono infarciti i testi di informatica). Usando una delle utility citate nel riquadro di queste pagine, vedremo di esaminare ciò che è successo al nostro dischetto.

A seconda dell'utility che possedete, posizionatevi ora sulla **traccia zero**.

Esplorando

Ciò che compare nel **primo settore** assoluto (numerato con zero), è il **nome** del sistema che ha formattato il dischetto stesso.

Ciò significa che se avete formattato con l'Ms-Dos Commodore, comparirà la parola Commodore; se con Epson, la parola Epson, e così via. Altri caratteri figurano prima e dopo il nome di identificazione. Per il momento, però, interessa solo sapere l'esistenza di questa "marchiatura".

Nella **seconda parte** del settore zero (ogni settore, è bene sottolinearlo, è formato da 512 byte, suddivisi in due blocchi da 256 ciascuno, oppure numerati da 0 a 512) è presente un messaggio il cui significato (in lingua italiana, tedesca, inglese o altra) è simile a quello riportato in figura 1 o, in definitiva, al seguente:

"Questo non è un disco di sistema; provvedi a sostituirlo e premi un tasto".

Ricordate, infatti, che abbiamo formattato il disco in modo "semplice", senza riportare i file di sistema? Bene, se tentassimo di accendere il computer inserendo il disco che abbiamo appena formattato, comparirà esattamente quel messaggio. Ne approfittiamo per suggerire una prima, semplice applicazione. Servendosi di uno dei Tool indicati in queste pagine, modificate il messaggio facendo in modo, ovviamente, di non occupare uno spazio maggiore di quello ad esso destinato (e restando sempre nel settore zero, per carità).

Ad esempio, fate in modo che sul monitor compaia la videata riportata in figura 2 e provvedete ad aggiornare il messaggio; con **PCTools**, ad esempio, il comando è impartito dal tasto **F5**.

Spegnendo e riaccendendo, oppure premendo contemporaneamente i tasti **Control**, **Alt** e **Delete**, vedrete la comparsa del nuovo messaggio. A che può servire una cosa del genere? Pensate ad un amico da prendere in giro, a patto che non sia lettore di Commodore Computer Club...

Favorite il vostro nome

Passiamo oltre; "rientriamo" nell'utility (PCTools, ad esempio) e andiamo ad esplorare gli altri settori del nostro dischetto. Nel settore **5**, a partire dal byte 0 (o nel settore **7** se adoperate un drive da 3.5) è presente il nome assegnato ad floppy (**Leucippo**) seguito da altri byte che, per il momento, non interessano. Se proseguite l'esplorazione dei settori (ve ne sono parecchie centinaia!) non c'è assolutamente nulla, o meglio vi sono circa 360000 byte di valore fisso, il cui codice cambia a seconda del Tool usato per formattare. Tali codici sono stati inseriti, automaticamente, al momento della formattazione.

Ora faremo un nuovo esperimento: "uscite" dal tool e formattate **nuovamente** lo stesso dischetto, ma in modo che vengano trasferiti i file di sistema e non il nome:

Format a:/s

Al termine dell'operazione (che può esser compiuta dallo stesso Tool, se questo lo consente) noterete che, esaminando la directory "normale" (**Dir** e return) non compare nulla; esaminandola, invece, con PCTools, noteremo la

Un programma prezioso

Con un computer, di solito, viene fornito un dischetto che contiene tutti i comandi del sistema operativo.

A volte, però, alcuni comandi sono piuttosto complessi, di difficile impostazione e non sempre se ne ricorda, agevolmente, la sintassi. Per questo motivo, pertanto, i dischetti forniti con la macchina contengono, oltre ai comandi di cui parliamo, altri programmi che, mediante un **menu** di facile comprensione, consentono di eseguire senza difficoltà particolare i compiti desiderati. Non si tratta, quindi, di nuovi comandi o, quanto meno, di una loro diversa forma sintattica. Ci troviamo, semplicemente, di fronte a comandi "vestiti a festa", circondati da cornici più o meno suggestive e da facilitazioni di editing più o meno valide.

Con i comandi in forma spartana, insomma, è sempre possibile attivare la più sofisticata e nascosta funzione del DOS, pur se con notevoli sacrifici e pagando un conto salatissimo (che può costringere allo spegnimento della macchina) in caso di errori.

Per esaminare con calma un dischetto può esser sufficiente il programma **Debug** che, opportunamente adoperato, permette di seguire i ragionamenti contenuti in queste pagine. Tuttavia Debug è piuttosto "duro" da usare, soprattuto per un principiante.

Fortunatamente sono in giro programmi di notevole affidabilità (e semplicità d'uso) che permettono di eseguire tutti i comandi del Dos e, allo stesso tempo, di esaminare, byte per byte, l'intero contenuto di un disco, floppy o hard che sia.

Non solo; sono anche dotati di particolari opzioni, assenti nel Dos "ufficiale", che risultano tuttavia indispensabili per operare in modo approfondito sul sistema stesso. I programmi di cui parliamo sono, principalmente, **Pctools** (della IBM) e le altrettanto note **Norton Utility**, entrambe in continua e costante evoluzione. Dal momento che risulta piuttosto semplice procurarsi tali programmi (anche le versioni meno recenti sono validissime per seguire le note di queste pagine) daremo per scontato il loro utilizzo, peraltro guidato da chiare ed esaurienti domande e/o menu di scelta.

Ai lettori che, purtroppo, non sono in possesso dei **Disk Editor** citati, non possiamo fare altro che consigliare di seguire con la massima attenzione il presente articolo, utilizzando, magari, i poco versatili programmi di esplorazione di un disco a loro disposizione.

I nomi, in codice

I file memorizzati su di un dischetto Ms-Dos possono essere allocati in una parte qualunque della superficie magnetica. La posizione esatta dipende da tantissimi fattori: preesistenza di altri file, file cancellati, presenza di settori danneggiati, eccetera.

Tuttavia, se ci riferiamo a un dischetto da 5 pollici e 1/4 (formattato a 360 K), a partire dal settore n. **5** saranno sempre memorizzate tutte le informazioni relative ai singoli file presenti, secondo un codice ben preciso. Questo è formato (lo ripetiamo: per ciascun file) da **32 byte** contigui, numerati da 0 a 31. Il loro significato è il seguente:

0 - 10: Nome del file, comprensivo di eventuale estensione.

11: attributo del file (**0** = normale; **1** = sola lettura; **2** = hidden file; **4** = file di sistema; **8** = nome del disco; **18** = sottodirectory; **32** = file non salvato; **64** = vuoto). E' possibile individuare la coesistenza di più valori (e quindi più attributi) fra loro.

12 - 21: non utilizzati (è pericolosa la loro manipolazione).

22 - 25: ora e data codificati secondo un ingegnoso algoritmo.

27 - 28: numero del cluster di partenza (puntatore).

28 - 31: dimensioni del file stesso.

In questa sede eseguiremo alcuni trucchetti manipolando, opportunamente, il byte degli attributi.

Altri byte possono esser manipolati, con conseguenze a volte... divertenti.

presenza di due file, dal nome **IBMBIO.COM** (di lunghezza compresa tra i 10 e i 18 Kbyte) e **IBMDOS.COM** (circa 30 K). Sono i famosi file nascosti (**hidden files**) di cui si parla altrove in queste pagine.

Vediamo che cosa è successo al disco, a parte la novità dei due file.

Anzitutto (a seconda del DOS che avete) potrebbe esser scomparso il messaggio relativo alla non idoneità del bootstrap (ricordate? era nel settore 0). Potrebbero esservi, al loro posto, altri codici che, appunto, comunicano al computer la disponibilità, sul disco inserito nel drive, dei file di sistema.

In fondo alla stessa "pagina" (sempre nel settore 0) sono comunque memorizzati alcuni messaggi che verranno evidenziati nel caso (malaugurato) in cui dovessero sorgere problemi con il disco stesso: smagnetizzazioni parziali, pasticci vari nella FAT (**F**ile **A**llocation **T**able, una parente stretta della **BAM** del 1541, lo ricordate?), assenza di Command.com e così via.

Nel settore 5 o nel settore 7 (drive da 5,25 oppure 3.5), al posto in cui c'era prima il nome del disco (Leucippo), non è rimasto nemmeno uno spazio vuoto (come ci saremmo aspettati), ma figurano i nomi dei due file di sistema che occupano, ciascuno, due gruppi di 16 byte, alcuni dei quali rappresentano i caratteri del nome (vedi riquadro).

Ciò è del tutto normale perchè, in un floppy di sistema, i primi due spazi disponibili per i nomi vengono proprio assegnati a Ibmbio e Ibmdos.

Continuando l'esplorazione dei settori ci accorgeremo che molti di questi sono riempiti di caratteri incomprensibili: sono i vari byte che costituiscono i file di sistema, cioè sono i programmi, in linguaggio macchina, che rappresentano il complesso di operazioni del Sistema Operativo. Ci accorgeremo, inoltre, che molti settori sono sostanzialmente identici tra loro, ma per ora questo particolare non interessa.

Ora proveremo a formattare nuovamente il disco, ma chiederemo di inserire anche il nome...

Format a:/v/s

..(drive da 360K)...

Format a:/4/v/s

..(drive da 1.2 M), che sarà, stavolta, "**Santippo**".

Esaminando, ora, la traccia 5 ci accorgiamo che i primi due file sono ancora Ibmbio e Ibmdos (e ciascuno occupa ancora 32 byte, due righe da 16 caratteri ciascuna) e che il nome del disco non è posto subito dopo di loro (da 64 a 95) ma un paio di righe più in basso, da 96 a 127. Perchè sono stati "saltati" 32 byte?

Decimale ed esadecimale

Agli appassionati che hanno appena acquistato un computer non possiamo fare altro che suggerire di rintracciare un testo in cui venga spiegata la (ahimè, lunga e noiosa, ma fondamentale) trattazione sui sitemi di numerazione, in particolare sulla **conversione** di un numero da decimale a esadecimale e viceversa.

L'argomento, infatti, è uno dei primi da studiare nel campo dell'informatica.

Per non costringere i lettori "esperti" ad inutili ripetizioni, ci limitiamo a pubblicare un semplice programma (in Basic) in grado di visualizzare la tabella di conversione che, comunque, viene qui riportata.

Le applicazioni suggerite in queste pagine, infatti, si riferiscono a videate in cui compaiono codici formati da due caratteri (alfabetici e numerici) che possono, in apparenza, risultare misteriosi. Vedremo un paio di esempi, con ovvio riferimento alla tabella; in questa, la colonna di sinistra si riferisce al primo dei due caratteri; la prima riga, invece, al secondo.

L'incrocjo tra la riga e la colonna individuerà il numero decimale (e viceversa).

Il numero (esa) **A8** si riferisce al valore (decimale) **168**. Analogamente a **3F** corrisponde **63**, a EF, 239 e così via. Al contrario, al numero decimale 92 corrisponde 5C eccetera. Vi sono casi in cui è possibile generare confusione: a che cosa corrisponde 38? a 56 (se 38 è un numero esadecimale); a 26 se, al contrario, 38 è un numero decimale.

Per evitare errori, quindi, nei testi di informatica viene sempre posto, accanto al valore, il sistema di numerazione di riferimento.

Esempio: 45 (esa); 128 (dec); A5 (esa); in quest'ultimo caso, magari, la specificazine è superflua, grazie alla presenza dell'inconfondibile carattere alfabetico.

```
100 REM CONVERSIONE DA SISTEMA DECIMALE A ESADECIMALE
110 :
120 X1$ = "0123456789ABCDEF": PRINT SPC(6);
130 FOR I=1 TO 16: PRINT MID$ (X1$, I, 1); "   ";: REM 3 spazi
140 NEXT: PRINT: PRINT
150 I=1: J=0: GOSUB 210: FOR K=0 TO 255
160 X2$ = RIGHT$ ("...." + STR$ (K), 4)
170 PRINT X2$;
180 J=J+1: IF J=16 THEN J=0: PRINT:GOSUB 210
190 NEXT K: PRINT
200 PRINT"Tabella di conversione decimale / esadecimale": END
210 PRINT MID$ (X1$, I, 1); CHR$(32); CHR$(32);: I=I+1
220 RETURN: END
```

Il brevissimo programma Basic

Il disco di sistema

Abbiamo detto che i due file Ibmbio e Ibmdos sono file di sistema; non sono, però, gli **unici**. Affinchè un dischetto sia realmente Bootable (che consenta, se inserito nel drive, di attivare il computer al momento della sua accensione) deve contenere anche un altro file, il ben noto(!) **Command.Com**; se questo dovesse risultare assente, al momento dell'eventuale boot comparirebbe la richiesta "Disco non di sistema; sostituiscilo e premi Return", o simili.

E' bene sottolineare che al momento della formattazione (con la sintassi: **/s**), alcune versioni di Ms-Dos **non** trascrivono anche Command.com, ma **solo** Ibmbio e Ibmdos; all'utente è affidato il compito di copiare, a parte, Command.com.

E' bene anche precisare che sono in circolazione versioni di Ibmbio e Ibmdos **incompatibili** sia tra loro sia con alcune delle varie versioni (ancora più numerose) di Command.com.

Consigliamo caldamente, quindi, di realizzare un disco di sistema utilizzando **esclusivamente** il comando di formattazione del vostro computer ed evitando (per scongiurare il pericolo del blocco totale del sistema) di prelevare i tre file da "fonti" diverse. Se, quindi, copiamo (dal disco sistema fornitoci con il computer) il file Command.

Com (a meno che l'utility di formattazione non l'abbia già fatto), ci accorgeremo che le informazioni ad esso relative verranno allocate proprio nei 32 byte lasciati vuoti (da 64 a 95 del settore 5). Ricordiamo, ora, che un disco bootable, oltre ai tre file citati, contiene solitamente (nella configurazione minima) sia il file specifico relativo alla tastiera collegata (**Key**bit, Keybfr, Keybgr e così via) sia il file Autoexec.bat che contiene (almeno) il comando di caricamento del file di tastiera. Con versioni più recenti del Dos il file della tastiera è uno solo e richiede una particolare sintassi per la sua corretta attivazione. In dischi di sistema più complessi compaiono numerosi altri file; per il momento ci limiteremo a studiare ciò che accade al floppy nella configurazione "minima" possibile.

Ora creeremo il file **Autoexec.bat** con un qualsiasi text editor (se lo possedete), oppure con l'obsoleto comando **Edlin** (se avete voglia di leggere le istruzioni sul vostro manuale) oppure digitando **alla lettera** quanto segue, senza curarvi di ciò che fate ([R] = premi il tasto Return) assicurandovi, però, che nel drive sia inserito il disco cavia:

Copy con: autoexec.bat [R]
keybit [R]
echo io sono il tuo computer [R]

A questo punto premete i tasti **Control** e "**Z**" e, subito dopo, Return. Il drive ronzerà un po' e, ad una verifica (**Dir**) vedremo che, in effetti, sul disco è presente il file di nome Autoexec.bat.

Copiate ora il file relativo alla vostra tastiera (supponendo che sia Keybit.com) in modo che, alla fine, la directory contenga:

Command.com
Autoexec.bat
Keybit.com

Resettando il computer (Control, Alt, Delete) il sistema dovrebbe partire dopo aver caricato, automaticamente, Keybit.com e visualizzato il messaggio di saluto.

Perchè mai, ci chiediamo legittimamente, viene lanciato proprio il file Autoexec.bat, e non altri?

Automatismo controllato

Descriveremo ora un altro trucchetto (idoneo a far impazzire l'amico di prima) che ci consentirà, tra l'altro, di approfondire la conoscenza di alcuni meccanismi di Ms-Dos.

Con un programma di utility (continueremo a riferirci a PCTools), ed esploran-

```
      0   1   2   3   4   5   6   7   8   9   a   b   c   d   e   f

0   .. 0.. 1.. 2.. 3.. 4.. 5.. 6.. 7.. 8.. 9. 10. 11. 12. 13. 14. 15
1   . 16. 17. 18. 19. 20. 21. 22. 23. 24. 25. 26. 27. 28. 29. 30. 31
2   . 32. 33. 34. 35. 36. 37. 38. 39. 40. 41. 42. 43. 44. 45. 46. 47
3   . 48. 49. 50. 51. 52. 53. 54. 55. 56. 57. 58. 59. 60. 61. 62. 63
4   . 64. 65. 66. 67. 68. 69. 70. 71. 72. 73. 74. 75. 76. 77. 78. 79
5   . 80. 81. 82. 83. 84. 85. 86. 87. 88. 89. 90. 91. 92. 93. 94. 95
6   . 96. 97. 98. 99 100 101 102 103 104 105 106 107 108 109 110 111
7    112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127
8    128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143
9    144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159
a    160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175
b    176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191
c    192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207
d    208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223
e    224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239
f    240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255

Tabella di conversione decimale / esadecimale
```

Tabella di conversione decimale / esadecimale da 0 a 256

Caricat, puntat, floppy!

Un floppy disk da 5 pollici 1/4, formattato a 360 kbyte, è costituito, appunto, da circa 368 mila byte, molti dei quali a disposizione dell'utente.

Per raccapezzarsi tra questa miriade di potenziali informazioni, il sistema operativo suddivide il disco non solo in due zone (faccia **superiore** ed **inferiore** del floppy stesso) ma in tracce e settori. Le **tracce** sono di forma circolare e sono 40 per ciascuna faccia (numerate, quasi sempre, da **0** a **39**); ognuna di queste è formata da 9 **settori** contigui contenenti, ciascuno, 512 byte.

In totale, pertanto, risultano...

512 x 9 x 40 x 2 = 368640

...byte, detti, in gergo, 360 Kbyte.

Per individuare un byte, fra tutti gli altri, sarebbe scomodo numerarli uno per uno. Si preferisce, quindi, indicare una "zona" che racchiude alcune centinaia (migliaia) di byte e riferirsi ad essa. L'indice di cui parliamo, dal momento che viene indicato da un numero, si dice **puntato** da quel numero. Questo, più propriamente, prende quindi il nome di **puntatore**.

Il concetto di "puntatore" è ricorrente nell'informatica.

Con tale termine si intende, di solito, un numero che indica la posizione in cui è presente l'**inizio** del guppo di byte che costituiscono l'informazione cercata. Se, ad esempio, un certo byte rappresenta un puntatore, e questo vale 35, la prima informazione utile sarà presente a partire dal 35-mo dato. Nel caso specifico, se un puntatore, su disco, indica la posizione del primo byte di un file ivi memorizzato, il numero 35 può indicare, a seconda dell'utility usata, la traccia (o il settore) desiderato. Alcune utility utilizzano termini diversi (**cluster, cilindro, traccia, settore, settore relativo, settore assoluto, n. kbyte, n. buffer** e così via).

Riferendoci al sistema di puntatori utilizzato dal programma **PCTools**, possiamo dire che un disco è suddiviso in clusters; ognuno di questi in 4 settori che, a due a due, contengono 256 byte ciascuno.

Per determinare il settore in cui è memorizzato il primo byte del file "puntato" dal puntatore 27 - 28 (vedi riquadro sulla codifica del nome dei file), il calcolo è semplicissimo:

N. settore = n. cluster x 2 + 8

Inversamente:

N. cluster = (n. settore - 8) / 2

Se, ad esempio, il puntatore è 4B (esadecimale, cioè 75 in decimale), il settore di partenza è:

75 x 2 + 8 = 158 (dec)

Selezionando il settore n. 158, appunto, rintracceremo l'inzio del gruppo di informazioni richieste.

Nel caso di un disco rigido (ad esempio, da 20 megabyte), la formula è diversa:

N. settore = n. cluster x 4 + 107

Inversamente...

N. cluster = (n. settore - 107) / 4

Coloro che posseggono il drive da 1.2 mega possono agevolmente individuare la corrispondente relazione matematica che lega tra loro cluster e settori; basterà che ci seguano nei nostri esperimenti.

do il settore 5 (sempre per un dischetto da 5.25), ci accorgiamo che il file Command.com possiede il puntatore di cluster 2F (47 dec). Con la formula prima vista ci posizioneremo quindi sul settore **102** (= 47 x 2 + 8). All'inizio, come ovvio, vi saranno codici incomprensibili; basterà spostarsi sui settori da 105 a 107 per notare, però, la posizione dei vari **messaggi** che, durante l'uso del Dos, vengono visualizzati. Ecco, dunque, perchè è possibile farli apparire in tutte le lingue: basta modificare i codici ad essi relativi!

Ma non basta; nel settore 111 compaiono i messaggi di Copyright (magari da personalizzare a piacimento); ciò che ora interessa, però, è il settore 112 in cui, dal byte 152 a 163, è indicato il nome che deve possedere il file di bootstrap, completo di suffisso (cioè: Autoexec.bat).

Sfruttando le potenzialità del Tool adoperato, modificate il nome in "**Giovanni.Bat**" (o come vi pare, purchè abbia, comunque, la lunghezza di otto caratteri, oltre al suffisso). Alterando questa parte del comando Command.Com, e resettando il computer, vi accorgerete che questo non attiva più Autoexec.bat, proprio perchè, ora, il nome ricercato è Giovanni.bat. Questo file, con un sistema simile a quello prima illustrato, provvederete ora a crearlo:

copy con:giovanni.bat
echo off
echo non credo che riuscirai mai
echo a capire da dove viene il
echo messaggio che vedi
keybit

...ricordandosi, come prima, di premere Control Z e Return per concludere.

Resettando il computer verrà ora visualizzata la sequenza di frasi prima digitate; il lavoro, però, non è ancora alla fine. E' necessario "occultare" il file Giovanni.Bat lasciando inalterato Autoexec.Bat, benchè, ormai, superfluo.

Entrate nel Tool e, come prima, visualizzate il settore 5. Se avete seguito alla lettera i nostri consigli, i 32 byte relativi a Giovanni.Bat dovrebbero essere quelli numerati da 192 a 223; il byte che rappresenta l'attributo è il **dodicesimo** (a partire dall'inizio); nel nostro caso è il byte n. 203 che contiene il codice **20** (esadecimale). Infatti, come si può notare dal riquadro, 20 è il codice di file "normale"; per renderlo "**hidden**" (invisibile, cioè, alla normale richiesta Dir) il suo attributo deve essere incrementato di 2. Fate in modo, quindi, di cambiare 20 in **22** ed esaminate, in seguito, la directory con il consueto comando Dir: dovrebbero apparire solo tre file (Command.com, Keybit.com, Autoexec.bat).

Il dischetto, così "conciato", parte ancora in autoboot ma, inspiegabilmente, visualizzerà un messaggio del tutto diverso da quello contenuto in Autoexec...

Alla vostra bontà è affidato il compito di limitare la pesantezza dei messaggi nascosti. Alla vostra intraprendenza, invece, la decisione di modificare gli altri byte che caratterizzano i file.

E vedere l'effetto che fa...

di Alessandro de Simone

...ED E' SUBITO PROSPETTIVA

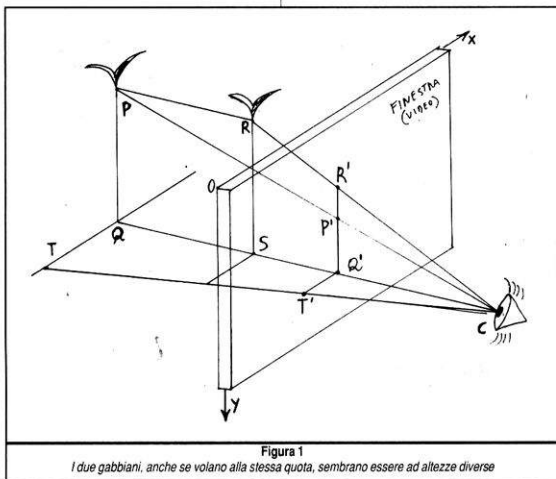
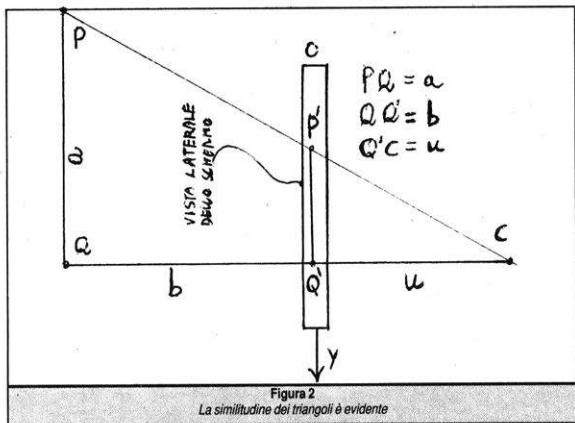
Le complesse regole per disegnare in prospettiva sono riducibili ad una sola istruzione, o meglio funzione, da elaborare due volte per ciascun punto considerato; sia che abbiate un C/64, un Amiga o un qualsiasi Ms-Dos compatibile

Supponiamo di essere in riva al mare (posizionati come in **figura 1**) e di osservare il volo di un gabbiano servendoci di un unico occhio puntiforme, anche se un po' bovino, chiamato **C**; il volatile, con tutto il rispetto che nutriamo per la natura, verrà schematizzato come un punto, denominato **P**.

Dal momento che siamo sulla riva, possiamo congiungere idealmente il nostro occhio C con l'estremità inferiore della linea verticale condotta dal gabbiano sulla superficie del mare (punto **Q**). L'uccello, quindi, sarà situato ad un'altezza **PQ** che rappresenta la quota reale alla quale è posizionato. Il segmento **CT** rappresenta la linea orizzontale che, partendo da C, attraversa, in maniera perfettamente perpendicolare, un'ipotetica **finestra** (rappresentata con il suo spessore) posizionataci davanti.

Possiamo notare facilmente che la linea CP la interseca in un punto **P'** così come la linea CQ la incontra nel punto **Q'**. Il segmento **P'Q'** rappresenterà una quota che possiamo definire **illusoria** (di solito pochi centimetri contro le varie decine di metri reali) che indica, in un sistema grafico chiamato **prospettiva**, la distanza relativa, **pseudo** proporzionale, tra altri elementi appartenenti alla stessa rappresentazione grafica. Ciò significa, ad esempio, che se il gabbiano vola a 30 metri di altezza, la distanza P'Q' viene ridotta ad un paio di centimetri se, ovviamente, lo schermo dista **CQ'** dal nostro occhio e la finestra dista **QQ'**

Figura 1
I due gabbiani, anche se volano alla stessa quota, sembrano essere ad altezze diverse

Figura 2
La similitudine dei triangoli è evidente

dalla verticale condotta dal gabbiano. Non tutto ciò che si trova a 30 metri dal suolo, però, verrà rappresentato da un segmento di eguale lunghezza.

Se un gabbiano **R**, infatti, si trova alla stessa altezza di P (notare il segmento verticale **RS** eguale a PQ), ma più vicino al nostro occhio (notare **CS** minore di CQ) la sua rapprentazione, limitatamente all'altezza dal suolo, sarà indicata dal segmento **R'Q'**, visibilmente maggiore di **P'Q'**.

Del resto non è un mistero per nessuno che oggetti piccoli, ma vicini all'osservatore, appaiono più grandi di altri oggetti, di dimensioni ragguardevoli, posti più lontano.

Un esempio per tutti è rappresentato dalla Luna, che appare notevolmente più grande delle stelle, che sappiamo essere immense. Nella figura 1 compaiono altre lettere, di cui ci occuperemo tra breve.

Matematica

Vediamo ora di esprimere matematicamente quanto osservato finora; niente paura, ragazzi

Sappiamo benissimo che la matematica non ha grandi sostenitori e vi assicuriamo che ci limiteremo a poche operazioni elementari.

Per meglio osservare lo schema, quindi, riferiamoci ora alla **figura 2**, che non è altro se non la figura 1, priva di gabbiani e vista di profilo.

Balza subito all'occhio che ci troviamo di fronte a due triangoli **rettangoli** (le altezze, per definizione, generano un angolo retto) e **simili** tra loro: il primo è **CPQ** ed il secondo **CP'Q'**. Dalle note regole della similitudine di elementi omologhi (roba da **scuola media**, ragazzi!) risulta la proporzionalità seguente:

PQ : P'Q' = QC : Q'C

Ne consegue che (un medio è dato dal prodotto degli estremi diviso l'altro "medio")...

P'Q' = (PQ * Q'C) / QC

Poichè, però, risulta che...

QC = QQ' + Q'C

...possiamo anche scrivere...

P'Q' = (PQ * Q'C) / (QQ' + Q'C)

Per semplificare la simbologia usata, denominiamo con lettere minuscole i segmenti esaminati:

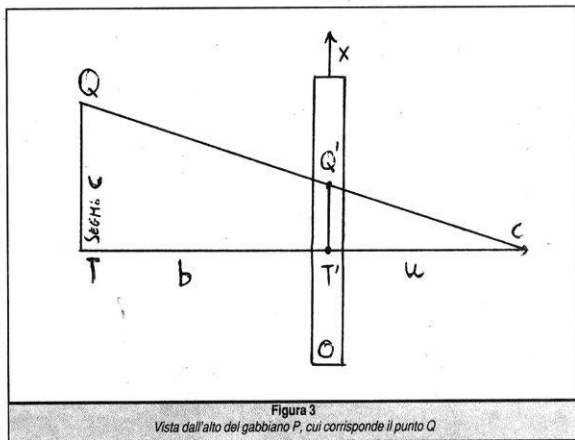
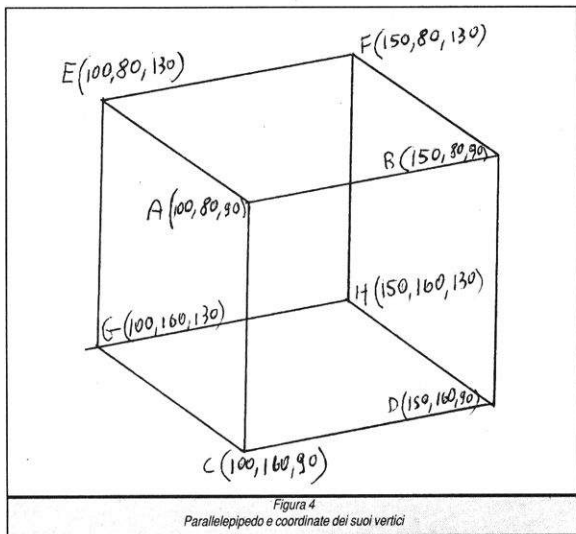
a = PQ
u= Q'C
b = QQ'

Il segmento incognito **P'Q'**, pertanto, potrà ora essere calcolato con la massima semplicità:

P'Q' = (a * u) / (u + b)

Ciò significa, in altre parole, che conoscendo la distanza dalla finestra (video...) del nostro occhio (costante **u**), l'altezza (**a**) di un punto e la sua distanza dallo schermo misurata secondo la proiezione verticale (**b**), è possibile determinare, con **una sola istruzione**, l'altezza virtuale del punto P sul nostro schermo video (segmento P'Q').

Figura 3
Vista dall'alto del gabbiano P, cui corrisponde il punto Q

Figura 4
Parallelepipedo e coordinate dei suoi vertici

Orizzontalmente

Abbiamo detto che la figura 2 rappresenta la figura 1 vista di profilo. E' ora il caso di ritornare alla figura 1, e di spiegare il significato degli altri simboli presenti.

Anzitutto c'è da notare che lo schermo viene identificato secondo due assi, **x** ed **y**, che non sono posizionati come abbiamo imparato a scuola, ma in accordo con la convenzione usata da tutti gli sviluppatori di linguaggi: l'asse x è positivo verso **destra**, l'asse y è positivo verso il **basso** (e non verso l'alto). Il punto **O** di origine delle coordinate, quindi, si trova nell'angolo in alto a sinistra dello schermo.

La **figura 3** rappresenta ancòra la figura 1, ma vista dall'alto: in questo caso l'asse x andrà verso l'alto (notare l'origine O posta in basso).

Le considerazioni sui triangoli rappresentati e sulle corrispondenti proporzionalità dovute alle loro similitudini, permetterà di determinare la **componente orizzontale** del gabbiano spostato verso destra (rispetto alla linea CT) della quantità **QT**; per un eventuale spostamento a sinistra varranno considerazioni pressochè identiche.

Siamo quindi giunti alla conclusione:

Per determinare le due componenti, orizzontale e verticale, di un qualsiasi punto P, è sufficiente elaborare l'unica relazione matematica... P'Q' = (a * u) / (u + b)

...che, a seconda dei casi, conterrà, oltre alla costante **u**, le due variabili **a** e **b** che assumeranno valori diversi a seconda della posizione "reale" del punto P nello spazio.

Infatti, dalla figura 3 si ha...

Q'T = (c * u) / (u + b)

...in cui il denominatore è identico a quello della precedente relazione ed al posto di **a** troviamo **c**.

E tutto, finalmente, collima: la prospettiva, infatti, è una rappresentazione **bidimensionale** di una realtà **tridimensionale**.

Ciò significa che da un punto, dotato di tre coordinate **x**, **y** e **z**, bisogna ricavare due sole coordinate (**x** ed **y**) dal momento che, su un foglio di carta (o su di uno schermo video) si hanno a disposizione solo **due** assi di riferimento. Si noti che, nelle "due" formule, la distanza che rappresenta la somma dei segmenti b ed u è sempre la stessa.

Ciò che cambia è il segmento indicato con **a** che assumerà i valori del caso a seconda se si considera la figura 2 come "vista" laterale o dall'alto della figura 1.

Dal punto allo spazio

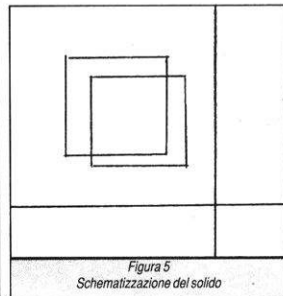
Le figure 2 e 3 si riferiscono, in effetti, alla rappresentazione di un solo punto.

Nel caso di un **rettangolo**, dotato di quattro punti, la figura si complicherebbe troppo per essere schematizzata; figurarsi, poi, se consideriamo un parallelepipedo; ma tentiamo lo stesso...

In **figura 4** ci siamo limitati ad indicare un parallelepipedo, dotato di **8** vertici, contraddistinti da altrettante coordinate. Si noti che i vertici appartenenti a ciascuna faccia (ciascuna parallela ad uno dei piani **xz**, **xy** oppure **yz**) hanno una coordinata in comune: la faccia ABCD dista 90 dal piano xy; la faccia AEGC dista 100 dal piano yz; la faccia ABFE dista 80 dal piano xz; e così via per le altre tre facce contraddistinte, a seconda dei casi, da identica coordinata x, y oppure z.

Se elaboriamo le "due" formule prima viste per ciascun punto, saremo in grado di tracciare la vista prospettica del nostro parallelepipedo.

Cose che, in pratica, svolgono i programmi presentati.

Figura 5
Schematizzazione del solido

La versione Basic

Il listato, scritto in **Gw-Basic**, gira in modo egregio sul **C/64** (dotato di emulatore Gw-Basic), in **Quick Basic** (potente interprete - compilatore Microsoft) ed **AmigaBasic**; in quest'ultimo caso è indispensabile eliminare (riga **170**) il comando **Screen 1**, che in Amigabasic non solo è superfluo (Amiga lavora sempre in modalità grafica), ma genera un syntax error.

Se avete digitato il tutto correttamente, date il Run e rispondete con **50** alla domanda *"Valore di u?"*. Questo valore rappresenta, appunto, la distanza u del nostro occhio dallo schermo (vedi figure 1, 2, 3). Subito dopo, premendo i tasti **+** (= allontana), **-** (= avvicina), **z** (= sinistra), **x** (= destra), **c** (= abbassa), **d** (= innalza) si potrà notare un parallelepipedo, schematizzato dalle due facce anteriore e posteriore, muoversi sullo schermo secondo le regole della prospettiva centrale. Gli assi x ed y saranno sempre presenti sul video per consentire di valutare il posizionamento del solido in movimento. Premendo il tasto **1** il programma termina.

Trucchetti Basic

Il programma, in effetti, **non** disegna i due rettangoli mediante la tracciatura di otto segmenti. Per evitare complicazioni, infatti, si è utilizzata l'istruzione **Line** con il parametro **,,B** (disegna un rettangolo) e **,,BF** (disegna un rettangolo "pieno") dopo aver fornito le coordinate della **diagonale** del rettangolo desiderato.

Se, ad esempio, vogliamo disegnare un solido formato da trapezi, saremo costretti ad utilizzare tutte le linee Basic "nascoste" dalle Rem (vedi parte terminale del listato) al prezzo, inutile dirlo, di una maggiore lentezza operativa.

Allo stesso modo (vedi **figura 5**) i vertici delle due facce anteriore e posteriore non sono uniti tra loro. Chi lo volesse, ovviamente, può introdurre le linee necessarie per effettuare l'elaborazione grafica dovuta.

In effetti non è nemmeno indispensabile elaborare i punti X2, X3, X6, X7, Y2, Y3, Y6, Y7 (vedi righe **320 - 360**) dal momento che non vengono utilizzati nel corso delle operazioni. Sono stati inseriti, tuttavia, per facilitare la modifica del programma da parte di quei lettori che volessero tracciare figure comunque irregolari.

Per quanto riguarda il riempimento "colorato", si noti la forma sintattica Line... ,,BF e la "duale" **Line... ,0,BF** che, rispettivamente, riempono e vuotano un rettangolo.

La sintassi necessaria per definire una funzione dovrebbe esser nota; ci limiteremo a ricordare che la funzione **FnPros** (riga 160) richiede, in ingresso, due parametri indicati, simbolicamente, da **A** e **B**. La variabile **U**, presente nella formula, conterrà il valore ad essa associato al momento della "chiamata".

Così, ad esempio, quando la riga 320 invoca...

X1 = FnPros (Xa, Z1)

...questa passerà i valori **Xa** e **Z1**, rispettivamente, ad **A** e **B**. La variabile **U** continuerà a valere, invece, quanto digitato in fase di Input. **X1**, quindi, conterrà il valore elaborato nel modo descritto.

Si precisa che possono verificarsi malfunzionamenti a seconda del computer usato, della scheda grafica montata e delle coordinate di riferimento selezionate (cioè i valori 359 e 169 presenti nelle istruzioni Line). Gli assi di riferimento, tracciati con coordinate massime 719 e 359, si riferiscono ad un computer Ms-Dos dotato di scheda Hercules.

Un'interruzione di programma si verifica se il denominatore della funzione (**u + b**) diventa nullo (division by zero error). Potevamo certo ricorrere ad un ciclo **If... Then** disponibile, anche all'interno di **Def FN**, nei moderni linguaggi Basic, ma abbiamo preferito non avvalercene per consentire la compatibilità completa con il **C/64** dotato di **emulatore Gw-Basic**.

Con un po' di pratica, comunque, sarà possibile individuare i valori esatti per realizzare un sistema di rappresentazione assonometrico rigorosamente "centrale".

La versione Turbo Pascal

In **Turbo Pascal** (originale **Borland**) la faccenda si complica leggermente a causa delle esigenze di questo meraviglioso compilatore.

Anzitutto non è disponibile una banale istruzione **Screen**, come in Basic, per "entrare" in grafica.

E' infatti indispensabile una procedura (presa pari pari da un demo della Borland) che attivi la pagina grafica a seconda dello schermo adoperato.

La procedura di cui parliamo (e che deve esser presente in **qualsiasi** programma T. Pascal che richieda schermate grafiche) è indicata con **Inizializza**. Si noti che il terzo parametro di **initgraph** rappresenta il percorso che il compilatore deve seguire per individuare le routine da attivare. Nel caso questo venga omesso (o meglio, sostituito da una coppia di apici) verrà considerato, come di default, il disco "attuale". Ciò significa che se lanciate il T. Pascal "dal" drive **a:**, al momento del Run potrebbero non esser "visibili" le routine richieste. Si consiglia, onde evitare equivoci, di indicare sempre il percorso necessario, come appunto indicato nel listato; o di lanciare T. Pascal dopo essersi posizionati nella sua subdirectory.

Per quanto riguarda il resto, nulla da dire: il programma è la traduzione pedissequa di quello Basic, a parte qualche procedura posta all'inizio, come è giusto che sia.

Attenzione al costrutto **If... Then** che deve sempre contenere un **Begin... End**, pena il mancato riconoscimento di alcune assegnazioni.

Da notare, inoltre, la **non** necessaria indentazione **nè** l'isolamento delle istruzioni **begin... end** (o altre). Si osservi, a questo proposito, le procedure **Tasto**, **Assi**; parte della funzione **Pros** e così via. Il compilatore, insomma, non ha bisogno di schematizzazioni; nè noialtri, del resto, cadiamo nel pallone se varie istruzioni sono presenti su di uno stesso rigo.

Ricordiamo che la funzione **Rectangle** (Line, nella versione T. Pascal è stata del tutto trascurata: datevi da fare per inserirla al posto giusto se lo desiderate...) richiede parametri in notazione intera.

Si richiede, pertanto, il trattamento di **variabili intere** oppure, per una migliore precisione, l'elaborazione di **variabili reali** e la successiva "traduzione" (grazie a **round**) in variabili intere.

Attenzione a **ClrScrn** e **Cleardevice** che, rispettivamente, cancellano lo schermo testo e quello grafico (con il Basic l'istruzione è **Clr** in entrambi i casi).

```
100 CLS: PRINT "GW - Basic Introduzione alla prospettiva"
110 PRINT "by Alessandro de Simone": PRINT
120 PRINT "Tasti + (allontana) - (avvicina)"
130 PRINT "Tasti x (destra) z (sinistra)"
140 PRINT "Tasti d (sopra) c (sotto)"
150 PRINT: PRINT "(premi 1 per finire)": PRINT
160 DEF FNPROS (A, B) = U * A / (U + B)
170 INPUT "Valore di U (1 - 100)"; U: SCREEN 1: CLS: GOSUB 400
180 REM valori solo per schermo 720 x 340
190 REM 359=720/2; 169=360/2 (compreso il punto 0)
200 XA = 10: XB = 150: XC = 10: XD = 150: Z1 = 220: Z2 = 240
210 YA = 80: YB = 80: YC = 160: YD = 160
220 GOSUB 560: IF A$ = "" THEN GOTO 220
230 IF A$ = "+" THEN Z1 = Z1 + 20: Z2 = Z2 + 20: GOTO 310
240 IF A$ = "-" THEN Z1 = Z1 - 20: Z2 = Z2 - 20: GOTO 310
250 IF A$ = "z" THEN XA = XA - 20: XB = XB - 20: XC = XC - 20: XD = XD - 20: GOTO 310
260 IF A$ = "x" THEN XA = XA + 20: XB = XB + 20: XC = XC + 20: XD = XD + 20: GOTO 310
270 IF A$ = "d" THEN YA = YA + 20: YB = YB + 20: YC = YC + 20: YD = YD + 20: GOTO 310
280 IF A$ = "c" THEN YA = YA - 20: YB = YB - 20: YC = YC - 20: YD = YD - 20: GOTO 310
290 IF A$ = "1" THEN SCREEN 0, 0, 0: CLS: END
300 GOTO 220
310 GOSUB 400: GOSUB 490: GOSUB 320: GOTO 220
320 X1 = FNPROS (XA, Z1): X2 = FNPROS (XB, Z1): X3 = FNPROS (XC, Z1): X4 = FNPROS (XD, Z1)
330 X5 = FNPROS (XA, Z2): X6 = FNPROS (XB, Z2): X7 = FNPROS (XC, Z2): X8 = FNPROS (XD, Z2)
340:
350 Y1 = FNPROS (YA, Z1): Y2 = FNPROS (YB, Z1): Y3 = FNPROS (YC, Z1): Y4 = FNPROS (YD, Z1)
360 Y5 = FNPROS (YA, Z2): Y6 = FNPROS (YB, Z2): Y7 = FNPROS (YC, Z2): Y8 = FNPROS (YD, Z2)
370 GOSUB 420: RETURN
380:
390 REM disegno assi:
400 LINE (359, 0) - (359, 359): LINE (0, 169) - (719, 169): RETURN
410:
420 LINE (359 + X1, 169 - Y1) - (359 + X4, 169 - Y4),, B
430 ' LINE (359 + X1, 169 - Y1) - (359 + X2, 169 - Y2): LINE (359 + X2, 169 - Y2) - (359 + X4, 169 - Y4)
440 ' LINE (359 + X4, 169 - Y4) - (359 + X3, 169 - Y3): LINE (359 + X3, 169 - Y3) - (359 + X1, 169 - Y1)
450 LINE (359 + X5, 169 - Y5) - (359 + X8, 169 - Y8),, BF
460 ' LINE (359 + X5, 169 - Y5) - (359 + X6, 169 - Y6): LINE (359 + X6, 169 - Y6) - (359 + X8, 169 - Y8)
470 ' LINE (359 + X8, 169 - Y8) - (359 + X7, 169 - Y7): LINE (359 + X7, 169 - Y7) - (359 + X5, 169 - Y5)
480 RETURN
490 LINE (359 + X1, 169 - Y1) - (359 + X4, 169 - Y4), 0, BF
500 ' LINE(359 + X1, 169 - Y1) - (359 + X2, 169 - Y2), 0: LINE (359 + X2, 169 - Y2) - (359 + X4, 169 - Y4), 0
510 ' LINE(359 + X4, 169 - Y4) - (359 + X3, 169 - Y3), 0: LINE (359 + X3, 169 - Y3) - (359 + X1, 169 - Y1), 0
520 LINE (359 + X5, 169 - Y5) - (359 + X8, 169 - Y8), 0, BF
530 ' LINE(359 + X5, 169 - Y5) - (359 + X6, 169 - Y6), 0: LINE (359 + X6, 169 - Y6) - (359 + X8, 169 - Y8), 0
540 ' LINE(359 + X8, 169 - Y8) - (359 + X7, 169 - Y7), 0: LINE (359 + X7, 169 - Y7) - (359 + X5, 169 - Y5), 0
550 RETURN
560 A$ = INKEY$: IF A$ = "" THEN GOTO 560
570 RETURN: END
```

La funzione **Def Fn**, in Basic, ha una sintassi del tutto simile a quella del T. Pascal ed assomiglia, per ciò che elabora ad una subroutine cui è possibile "passare" parametri da elaborare.

Xa, Xb... Yd rappresentano le coordinate dei vertici del parallelepipedo. Si notino le due sole coordinate di Z (z1 e z2).

Il disegno del solido viene schematiazzato grazie a due rettangoli "sfalsati" tra loro secondo le regole della prospettiva.

Si noti la sintassi, **,0,BF** (per cancellare un rettangolo pieno); **,,BF** (per "riempirlo" con il colore attuale); **,,B** (per tracciarne solo i contorni). Si noti, inoltre, la mancata elaborazione di numerose linee grazie alla presenza del simbolo apostrrofo (') corrispondente al comando REM.

Per come è scritto, il programma è totalmente compatibile con il **C/64** (dotato di emulatore Gw-Basic) con un qualsiasi **Ms-Dos compatibile** e con **Amigabasic** (a patto di eliminare le istruzioni **Screen**).

La procedura **Inizializza** è indispensabile, operando in T. Pascal, per attivare lo schermo grafico. Il compilatore ricerca le routine di cui necessita nel **path** (percorso) indicato nel terzo parametro di **Initgraph**. Nel nostro caso queste sono presenti nella directory **TP** del disco rigido (**C:**).

La funzione **Pros** deve restituire un valore **intero** per consentire la successiva elaborazione della procedura **rectangle**. Si noti la possibilità di effettuare salti (**If... Then**) nel caso in cui il denominatore risulti nullo e la trasformazione di un valore reale nel corrispondente intero grazie a **round**.

La tecnica della **programmazione strutturata** viene posta in evidenza dalle numerose **procedure**, una specie di subroutine del Basic, che sono in grado di snellire il lavoro del programmatore.

Qui, finalmente, inizia il programma vero e proprio con una schermata di presentazione del tutto simile a quella ottenibile con il Basic. Si noti **ClrScr** che cancella lo schermo testo.

Due cicli **Repeat** consentono di esaminare la pressione di un tasto. Il primo attende che venga abbassato un tasto qualunque (**keypressed**); il secondo, invece, dirotterà il flusso del programma a seconda di ciò che si digita, grazie alla procedura **tasto**.

```
program prospettiva;
    uses
    Crt, Graph;
procedure inizializza;
    var
    gr, gd : integer;
    begin gr := detect;
    initgraph (gr, gd, 'c:\tp');
        if graphresult < > grok then
            begin writeln ('Errore di apertura grafica'); halt (1)
            end;
end;
var xa, xb, xc, xd, x1, x2, x3, x4, x5, x6, x7, x8, z1, z2: integer;
var ya, yb, yc, yd, y1, y2, y3, y4, y5, y6, y7, y8, gr, gd: integer;
var u: real;
var s: string;
function PROS (a, b:integer):integer;
    begin if u+b < > 0 then begin; pros:= round ((u*a) / (u+b)); end;
    end;
procedure tasto; begin s:=readkey; end;
procedure cancella;
    begin setcolor(0);
        rectangle (359 + X1, 169 - Y1, 359 + X4, 169 - Y4);
        rectangle (359 + X5, 169 - Y5, 359 + X8, 169 - Y8);
    end;
procedure scrivi;
    begin setcolor(1);
        rectangle (359 + X1, 169 - Y1, 359 + X4, 169 - Y4);
        rectangle (359 + X5, 169 - Y5, 359 + X8, 169 - Y8);
    end;
procedure assi;
    begin; LINE (359, 0, 359, 359); LINE (0, 169, 719, 169);
    end;
begin; clrscr;
    writeln ('Turbo Pascal: Introduzione alla prospettiva');
    writeln ('by Alessandro de Simone');
    writeln ('Tasti + (allontana) - (avvicina)');
    writeln ('Tasti x (destra) z (sinistra)');
    writeln ('Tasti d (sopra) c (sotto)');
    writeln;
    writeln ('(per finire premerai 1)');
    writeln;
    write ('Valore di U (da 0 a 100)? '); readln (u);
    writeln ('(ora premi i tasti di controllo + - z x d c)');
repeat until keypressed;
inizializza; cleardevice;
XA := 10; XB := 150; XC := 10; XD := 150; Z1 := 220; Z2 := 240;
YA := 80; YB := 40; YC := 160; YD := 160;
assi;
repeat tasto; assi;
    If s = '+' then begin; Z1 := Z1 + 10; Z2 := Z2 + 10; end;
```

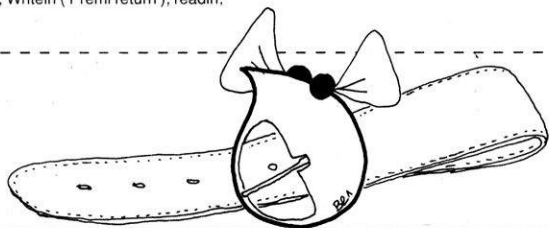
Le procedure si attivano semplicemente "invocandole" (**inizializza; assi;** eccetera). Per cancellare lo schermo grafico è necessario usare **cleardevice** dal momento che **ClrScr** è inefficace

I vari salti condizionati **If... Then** richiedono tassativamente le due istruzioni **begin** (che delimita la prima delle istruzioni da elaborare) ed **end** (da porre dopo l'ultima istruzione), pena il mancato riconoscimento di **tutte** le istruzioni intermedie.

Il costrutto If... Then del Basic, quindi, è del tutto diverso da quello del T. Pascal.

Con il **Basic**, come è noto, viene elaborata l'intera linea presente dopo **Then** nel caso la condizione sia riconosciuta vera.

Con il T. Pascal, invece, verrebbe elaborata **solo** l'istruzione successiva a Then!

```
IF s = '-' THEN begin; Z1 := Z1 - 10; Z2 := Z2 - 10; end;
    IF s = 'z' THEN
    begin;
        XA := XA - 10; XB := XB - 10; XC := XC - 10; XD := XD - 10;
    end;
IF s = 'x' THEN
    begin;
        XA := XA + 10; XB := XB + 10; XC := XC + 10; XD := XD + 10;
    end;
IF s = 'd' THEN
    begin;
        YA := YA + 10; YB := YB + 10; YC := YC + 10; YD := YD + 10;
    end;
IF s = 'c' THEN
    begin;
        YA := YA - 10; YB := YB - 10; YC := YC - 10; YD := YD - 10;
    end;
cancella;
X1 := PROS(XA, Z1); X2 := PROS(XB, Z1); X3 := PROS(XC, Z1); X4 := PROS(XD, Z1);
X5 := PROS(XA, Z2); X6 := PROS(XB, Z2); X7 := PROS(XC, Z2); X8 := PROS(XD, Z2);
Y1 := PROS(YA, Z1); Y2 := PROS(YB, Z1); Y3 := PROS(YC, Z1); Y4 := PROS(YD, Z1);
Y5 := PROS(YA, Z2); Y6 := PROS(YB, Z2); Y7 := PROS(YC, Z2); Y8 := PROS(YD, Z2);
scrivi;
until s= '1';
closegraph; Writeln ('Premi return'); readln;
end.
```

Per i più bravi

Ed eccoci ad un'ennesima **sfida**: i lettori più in gamba possono (anzi, devono...) elaborare una procedura in grado di:

- Adattare automaticamente la centralità della prospettiva alla dimensione dello schermo utilizzato; nulla di trascendentale. Ad esempio...

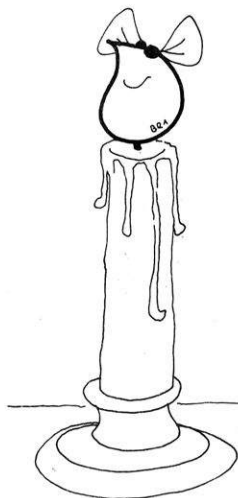
Input "Risoluzione orizzontale"; ro
Input "Risoluzione verticale"; rv

...una serie di If... Then successivi provvederanno a determinare gli opportuni riferimenti per le varie elaborazioni da compiere.

- Individuare la relazione matematica che tenga conto anche della **rotazione** di un segmento, rispetto ad uno qualsiasi dei tre assi, e tracci i suoi estremi sullo schermo, ovviamente in prospettiva; e qui (ahinoi) entra in gioco la **trigonometria**; insomma, solo roba per adulti, maggiorenni e vaccinati.

- Sofisticare la relazione precedente per consentire la tracciatura di un solido che, ruotando secondo uno (o più) assi, si sposti sullo schermo.

- Linee **nascoste**: che ne dite di cancellarle?

- Potremmo continuare, ma penso che basti per i prossimi dodici anni...

di Alessandro de Simone

A SCUOLA CON MICROSOFT

Parlando di linguaggi non si può fare a meno di parlare anche di didattica. L'interlocutore diretto, in questo campo, è certamente la scuola, con cui la Microsoft intende sviluppare un discorso costruttivo; e non solo a parole

Works è uno dei programmi su cui la Microsoft punta moltissimo per la diffusione della cultura informatica nelle scuole, soprattuto confortata dall'esempio offerto dal ministero della Pubblica Istruzione **francese**, che ha consentito la divulgazione del pacchetto integrato in diverse migliaia di esemplari presso le scuole pubbliche.

Per chi non lo sapesse, **Works** è un insieme di vari programmi che, pur operando indipendentemente l'uno dall'altro, consentono di scambiar dati e files per realizzare, appunto, ciò che in informatica viene definito un **ambiente integrato**.

E' presente uno **Spreadsheet** (foglio elettronico) per sviluppare calcoli concatenati, anche molto complessi, e rendere i risultati visibili sia in forma numerica, sia tabellare, sia grafica (diagrammi a torta, a barre eccetera). E' disponibile un potente **Word Processor** che, oltre a sofisticate funzioni di impaginazione, consente di "importare" tabelle numeriche sviluppate precedentemente dalla sezione Spreadsheet, e viceversa: i dati presenti in un gruppo di fatture, ad esempio, possono essere "letti" dallo Spreadsheet per effettuare calcoli e statistiche di ogni tipo; e magari reinserire i risultati nel testo riginario, scritto con la "sezione" word processor.

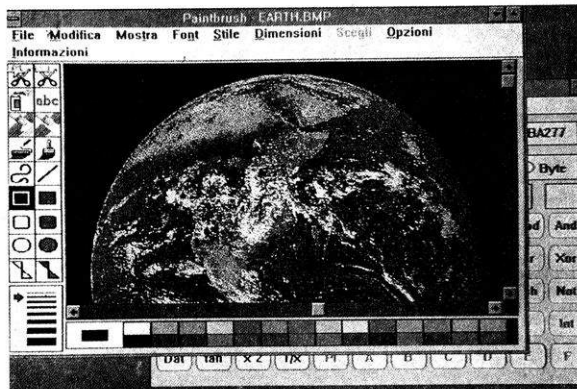
Il **Data Base** è il terzo, ma non ultimo, programma del pacchetto; consente di archiviare dati e di rintracciarli secondo "chiavi" di qualsiasi tipo.

Anche in questo caso, grazie alla integrazione, è possibile scambiare dati con gli altri programmi.

Se, ad esempio, state scrivendo una lettera circolare, è possibile stampare lettere personalizzate.

In queste, come indirizzo, viene riportato, uno per foglio, ognuno dei nominativi memorizzati precedentemente nel Data Base; e viceversa: potete scrivere una lettera ad un nuovo cliente e, alla fine, approfittare del fatto che l'indirizzo è presente nella lettera presente su video, per aggiornare il file dei clienti con il nuovo nominativo.

Altri brevi, ma utili, programmi fanno parte del pacchetto: una sofisticata **calcolatrice**, un **orologio** programmabile, in grado di dare la **sveglia**, ed un'**agenda** di appuntamenti; le tre utilities compaio-

no immediatamente sullo schermo, con la pressione di un tasto, qualunque cosa stiate facendo in quel momento.

La notizia più interessante, però, non è dovuta alle informazioni relative a Works, ma soprattutto al fatto che la Microsoft, periodicamente, offre il pacchetto a prezzi straordinari, pur se limitatamente a categorie di lavoratori della scuola, ed a configurazioni, ben definite.

Fino a poco tempo fa, ad esempio, un docente poteva procurarsi **Works 2** (in italiano) al prezzo di **L. 130 mila** (contro le 400 mila indicate nel listino ufficiale). **Ms - Mouse** (il versatile mouse Microsoft, da collegare alla porta seriale di un computer Ms-Dos) ed il programma grafico **PaintBrush** erano posti in offerta alla **metà** esatta del prezzo di listino.

E' bene ricordare ai ritardatari, comunque, che la Microsoft ripeterà, di tanto in tanto, interessanti offerte a docenti e scuole.

QuickPascal

Un Pascal un po' diverso dal solito è offerto agli operatori del mondo della scuola. Si tratta di un compilatore Pascal che, come punto di forza, offre la possibilità di evidenziare su schermo diverse categorie di termini in diversi colori. Le parole chiave del linguaggio, ad esempio, appaiono in rosso; gli identificatori sono in grigio, le stringhe in viola.

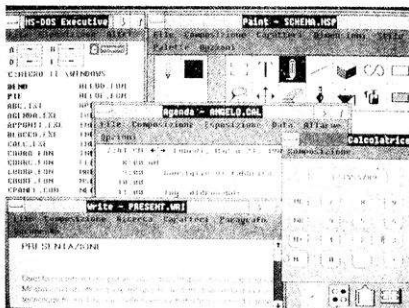
La diversa colorazione faciliterebbe, soprattutto nel caso di studenti... distratti, la ricerca degli errori e la conseguente, necessaria correzione; altri vantaggi sembrano derivare dall'originale idea di far apparire in vari colori i listati del compilatore. Non appena riceveremo il pacchetto per la prova, comunque, vi faremo sapere.

Anche in questo caso, trattandosi di un linguaggio in uso nelle scuole, Microsoft ha pensato bene di sviluppare una politica di prezzi particolarmente vantaggiosa per i docenti. Fino a poco tempo fa QuickPascal (come QuickBasic) era offerto esattamente alla metà del prezzo di listino; Windows 3, ambiente operativo nel quale si trovano a loro agio tutti i pacchetti Microsoft, era offerto a meno di un terzo del prezzo originale.

Programmi... all'ingrosso!

Microsoft, sensibile alle esigenze (soprattutto economiche) della scuola pubblica italiana, ha deciso di venire incontro alle necessità educative proponendo una formula di acquisto forse insolita, ma interessante: è infatti possibile procurarsi una confezione di prodotti applicativi a prezzo contenuto; questo viene ridotto ulteriormente acquistando la confezione **Lab-Pack** (10 licenze d'uso e 2 set di documentazione); con la confezione **University-Pack** (10 set di documentazione, 10 set di dischetti) il prezzo per singolo prodotto diventa minimo, ma questa opportunità è offerta alle sole facoltà universitarie.

Un esempio, per tutti: Works 5 (in italiano) viene ceduto a L. 240 mila (prodotto singolo); L. 400 mila (lab-Pack); un milione e 600 mila (University-Pack).

Molti altri prodotti, alcuni dei quali si possono usare anche in rete, vengono offerti a prezzi vantaggiosi.

L'intero gruppo di offerte occupa per intero le 4 pagine del depliant, da richiedere ad uno dei distributori Microsoft sparsi per l'Italia.

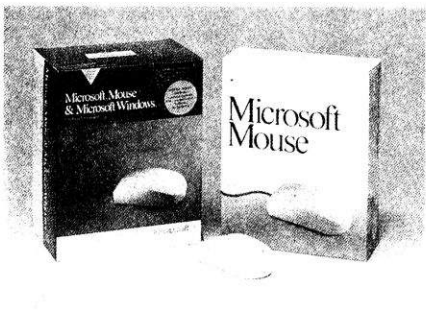
Giunge inoltre notizia che alcuni di questi, in concomitanza di particolari eventi (Fiere, manifestazioni culturali, eccetera) offrono, pur se per periodi limitati, offerte di vario tipo, sempre vantaggiose per l'utente finale.

Seminari Microsoft

Nel campo della formazione professionale (e la formazione dei docenti non è certo estranea a queto campo) Microsoft offre costantemente opportunità di rilievo, pur se limitatamente alle sedi di Roma e di Milano.

Nel mese di dicembre, ad esempio, nella capitale si è tenuto un corso di due giorni (Introduzione alla programmazione Windows 3) ed uno di un solo giorno (Introduzione al sistema operativo OS/2 2.0).

Nel **gennaio '91**, a Milano, nei giorni 17 e 18 sarà tenuto un corso dedicato all'Ambiente **Lan Manager 2.0**; nei tre giorni 28, 29 e 30, sempre a Milano, si introdurranno i corsisti alla programmazione in ambiente **OS/2** e in **Presentation Manager**.

Anche in questo caso consigliamo di contattare direttamente la Microsoft per ottenere informazioni sui prezzi (che ci assicurano contenuti) e sulle modalità di partecipazione ai corsi (Milano: tel. 02/21.07.201. Roma: tel. 06/68.92.802).

SUPERSCHEDA SUPERGRAFICA

Che cosa succede se decidiamo di abbandonare il bianco e nero per scoprire la verità policroma del nostro PC compatibile? (ed altre avventure)

Le schede grafiche

Purtroppo, nel mondo Ms-Dos, l'utenza paga pesantemente gli inspiegabili ritardi tecnologici verificatisi nello sviluppo del sistema.

Quando, infatti, videro la luce i primi **P**ersonal **C**omputer IBM (che, in seguito, avrebbero fatto la storia dell'informatica "personale") non si pensò nemmeno alla grafica, ma i suoi progettisti si limitarono ad offrire il modo testo nonostante fosse possibile, a quei tempi, produrre a basso costo circuiti elettronici in grado di gestire sia grafica che colore. Poco dopo, per fortuna, furono rese disponibili schede elettroniche, da inserire in uno degli slot del computer, che consentivano di "uscire" anche in alta risoluzione senza rinunciare, ovviamente, alla contemporanea presenza di caratteri di testo. Una delle schede più note fu certamente la **MDPA** (**M**onochrome **D**isplayer and **P**rinter **A**dapter) che offriva, oltre a numerosi modi testo (40 x 25; 40 x 43; 40 x 50; 80 x 25; 80 x 43; 80 x 50 con formato dei caratteri 8 x 8), anche due modalità grafiche: la 320 x 200 (in cui era possibile scrivere anche caratteri, in modo testo, in una griglia di 40 colonne x 25 righe) e quella 640 x 200 che, volendo, consentiva di scrivere caratteri Ascii in 80 colonne x 25 righe di testo. Per rendere più familiare l'argomento, ricordiamo che, in ambiente di programmazione Gw-Basic (o QuickBasic) la scheda MDPA viene riconosciuta dal comando **Screen 0** (solo modo testo); **Screen 1** (320 x 200); **Screen 2** (640 x 200).

Mi sono informato: siamo ormai alle soglie del 2000 e sembra (ripeto: sembra) che la civiltà dell'immagine, dominatrice assoluta del nostro pianeta, richieda anche il colore.

Guardo con malinconia il mio AT compatibile che offre uno schermo grigioide nel momento in cui visualizza una schermata graficoide in biancoide e neroide. Attorno al computer, invece, è presente la fiera del colore: dischetti policromi, la squadra dell'Italia coloratissima (la squadra, non l'Italia) fotografata in perfetta formazione, i ritratti delle mie ragazze, attualmente attive in time - sharing.

Perfino il tavolo, in imitazione legno, sembra coloratissimo in confronto alla schermata grafica. Prendo il libretto degli assegni ed esco per un fare il giro dei computervendoli. Entro nel primo negozio. Una famigliola attornia il ragazzino prodigio (da grande farà il dottore, lo si capisce da come il padre gli accarezza la testa) che con fare saputo manovra il joy di un Amiga.

Lo sceriffo di una banca vicina osserva la sezione Ninja dei videogames. Il titolare del negozio mi guarda con aria interrogativa.

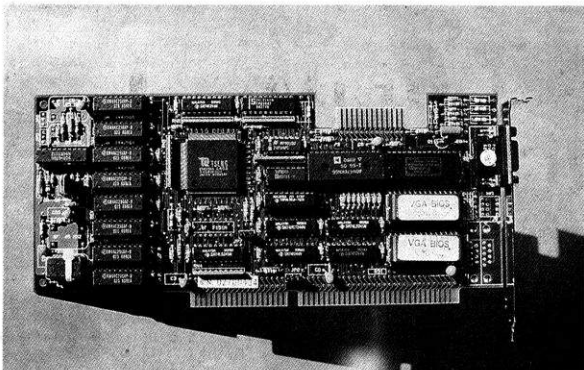
"Vorrei informazioni su una scheda Ega per il mio...".

"Che cooosa?" - esclama il titolare del negozio, appoggiandosi con le mani al bancone- *di questi tempi cerca ancòra una Ega?"*.

Mi sento osservato; la madre tenta di nascondere alla mia vista il ragazzino prodigio, lo sceriffo sfiora nervosamente il calcio della pistola mentre, senza farsene accorgere, divarica lievemente le gambe per avere una presa migliore in caso di conflitto a fuoco; lo ha visto fare in un telefilm e ripete pedissequamente.

"Il fatto è -tento tangenzialmente- *che i soldi per una VGA proprio non li ho."*

La scusa sembra convincente, la madre sembra rassicurata, lo sceriffo è certamente deluso per esser stato costretto a rinviare la tracciatura della prima tacca sul calcio della sua pistola.

"Ma caro signore -rassicura il computerista- *ormai i prezzi sono quasi livellati. Forse non sa che una scheda EGA (tra l'altro introvabile; io, almeno, non ce l'ho) costa ormai quanto una VGA?"*.

Si potrebbe obiettare che, chi possiede una scheda di questo tipo, non avrebbe motivo di lavorare in modalità di solo testo; analogamente, disponendo della modalità 640 x 200, non avrebbe senso operare limitandosi ad una griglia di soli 320 x 200 pixel.

Naturalmente la risposta alla giusta obiezione risiede nella diversa occupazione di memoria, nella possibiltà di far apparire caratteri più grandi, nel ricorso alla risoluzione più elevata solo in presenza di figure di una certa complessità: la velocità di elaborazione, come intuitivo, varia a seconda della selezione impostata. Un gradino più in alto troviamo la scheda di tipo **Hercules** (corrispondente a **Screen 3**). Questa, oltre alle potenzialità offerte dal tipo di scheda precedente, porta la risoluzione grafica dello schermo a 720 x 348, che è la risoluzione più elevata ottenibile... a basso costo, e risulta più elevata perfino di altre schede a colori.

Ed ora passiamo, appunto, alle schede a colori.

La più antica è la **CGA** (**C**olor **G**raphic **A**dapter) che offre 16 colori di testo (con Screen 0) e dispone di uno scrolling molto lento, spesso fastidioso. Con Screen 1 la risoluzione di una CGA è di 320 x 200 su uno schermo colorato in uno dei 16 colori disponibili, ed un testo (o grafica) in uno dei 4 disponibili.

Di solito una scheda CGA è in grado di simulare anche una MDPA (e, spesso, anche una Hercules). Settata in uno di quei modi, pertanto, offre le stesse caratteristiche prima esaminate.

Appena più moderna (ma scarsamente presente anch'essa perfino nelle configurazioni più economiche degli attuali computer Ms-Dos compatibili) è la **EGA** (**E**nhanced **G**raphic **A**dapter) che, presentata forse troppo tardi in un mercato informatico... coloratissimo, ha tuttavia tenuto banco per molto tempo.

Quasi tutte le schede EGA simulano la CGA, la MDA e, spesso, anche la Hercules. Se usate come vere e proprie EGA, queste offrono (ad esempio, con **Screen 7**) una risoluzione di 320 x 200 pixel con 16 colori; con **Screen 8** una risoluzione di 640 x 200 e con **Screen 9** si passa a 640 x 350 fino a 64 colori (a seconda della **Ram** installata sulla scheda).

Proprio così: il numero dei colori dipende, nelle schede EGA e successive, dalla quantità di memoria installata, di solito modulare di 64 KRam.

Dopo un tentativo non fortunato di varie schede SuperEga (che offrivano standard spesso insoliti), si è pervenuti

---

Non me ne sono accorto, ma mentre parlava, il bancone si è miracolosamente riempito di sette depliant e quattro confezioni di schede grafiche. Ne apre una, con lentezza, come si fa con i bambini quando si aprono le scatole dei cioccolatini. Ma una scheda è più buona, e non fa male alla pancia.

Estrae, infatti, un meraviglioso rettangolo ricco di connessioni dorate; circuiti neri dalle scintillanti zampe innumerevoli fanno bella mostra di sè. Una lucidissima piastrina in acciaio pone in evidenza il connettore per il monitor.

"Vuoi mettere?" dice con voce melliflua l'hardwarevendolo che improvvisamente di dà del tu. Non posso certo negarlo: è una SuperVga che occuperebbe in modo elegante uno slot a 16 bit del mio AT compatibile.

"Sì, d'accordo. Ma se sommiamo il prezzo del monitor a colori richiesto da questa bestia, a che cifra arrivo?".

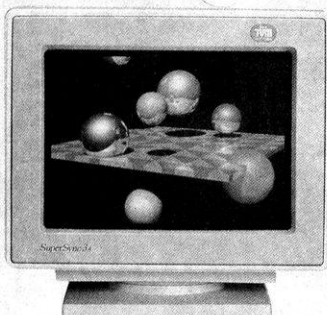
La risata fragorosa che segue alla mia ingenua osservazione (del tipo ma-guarda-un-po'-che tipi-vanno-in-giro-al-giorno-d'oggi) mi fa sentire un verme.

La madre di famiglia mi guarda con sguardo compassionevole; il padre di famigia sembra più comprensivo, è lui che paga sempre e comunque; lo sceriffo, invece, guarda il negoziante in attesa che gli faccia un cenno con il pollice verso, che per fortuna non arriva.

"Guardi -dice il computeraio, ritornando a misurare le distanze- *decida lei: comprare una SuperVga ed un monitor multisync, che le dureranno tutta la vita; oppure una EGA normale, con monitor corrispondente, da riporre in cantina fra un anno (al massimo) quando si vedrà costretto a comprare SuperVga e Multisync".*

Il dannato ha ragione; non glielo dico di certo, ma penso alla mia cantina in cui giacciono sia il monitor monocromatico che usai soltanto con il Vic 20, sia il monitor a colori che collegavo al C/64, e che ora è inservibile per il mio Ms-Dos. Quanti altri monitor porterò in cantina, prima di fermarmi?

Esco dal negozio tirando su il bavero; fa proprio freddo. Torno a casa e guardo il monitor grigioide, forse per l'utima volta.

Prendo una fotografia a caso, la giro e formo il numero di telefono. Ho proprio bisogno di distrarmi un po'.

allo standard **VGA** (**Vi**deo **G**raphics **Ar**ray) che è quello abitualmente offerto nelle configurazioni dei moderni personal computer.

Più attuale, grazie ad un certo successo commerciale, è certamente la **Super-Vga,** che viene offerta con configurazioni di Ram variabili tra **256** KRam, **512** KRam ed un **mega** di Ram.

Inutile dire che quest'ultima offre il meglio del meglio: risoluzione 1024 x 768 pixel in 256 colori: la qualità delle immagini dei demo, offerti su disco nella confezione di questi mostri, sono paragonabili a vere e proprie diapositive.

Non tutti i linguaggi commercializzati offrono istruzioni specifiche per tutte le risoluzioni ed il numero di colori offerti dalle schede poste di recente in commercio. Ma, ci assicurano, stanno provvedendo alla bisogna...

Bassa e alta risoluzione

Una stessa scheda grafica, soprattutto se moderna, offre una compatibilità spesso totale con altre schede di precedente produzione.

Alcuni programmi, caricati su computer che dispongono di tali schede, non sempre funzionano bene.

Vi sarà sicuramente capitato di osservare disallineamenti di messaggi, figure tracciate in modo parziale o sovrapposte ad altre, parti di schermo inspiegabilmente vuote, colori poco aderenti a ciò che ci si aspetta. Il motivo può esser addebitato a vari fattori.

Il primo è la vetustà del programma lanciato che, magari, scritto ai tempi di una CGA, viene fatto girare con i parametri di un'altra scheda. Un altro valido motivo è quello relativo ad una risoluzione diversa da quella richiesta dal programma. Altra causa può essere la presenza di una quantità di Ram insufficiente a gestire l'immagine generata.

Fortunatamente i programmi più seri tengono conto delle varie eventualità e si adattano automaticamente alla scheda installata oppure chiedono all'utente di indicarne con precisione le caratteristiche. I guai vengon fuori con i programmi scritti dagli appasionati e smanettoni.

Se alcuni controlli non vengono esplicitamente previsti ed eseguiti dal programmatore (durante la fase di scrittura del listato) il programma può presentare bugs se fatto girare su un computer dotato di scheda grafica con caratteristiche diverse da quelle riscontrabili sull'elaboratore sul quale è stato scritto.

In questi casi, disponendo del programma sorgente, è possibile intervenire ed apportare le correzioni del caso; altrimenti, se si possiede solo il file eseguibile, son dolori atroci...

Un'ultima avvertenza è d'obbligo: alcuni programmi professionali richiedono una configurazione di memoria minima che tende ad occupare quasi per intero i **640 KRam** standard.

Se non disponete di un'espansione di memoria, è realmente presente il pericolo di non poter utilizzare la scheda con tale programma!

Una qualsiasi modalità grafica, infatti, richiede che, al momento del boot del sistema, sia presente nel file **Config.Sys** il device relativo alla scheda stessa. A volte questa esigenza non è presente dal momento che la scheda stessa lancia un programma di gestione non appena viene data tensione.

In entrambi i casi, comunque, il device (=programma) tende ad occupare la memoria del computer, anche se non in misura considerevole. In questi casi, quindi, la Ram lasciata libera potrebbe non esser sufficiente a far girare il programma succhia-memoria. Il rimedio? Settare la nuova scheda grafica nella stessa modalità in cui operava la vecchia scheda che avete appena sostituito...

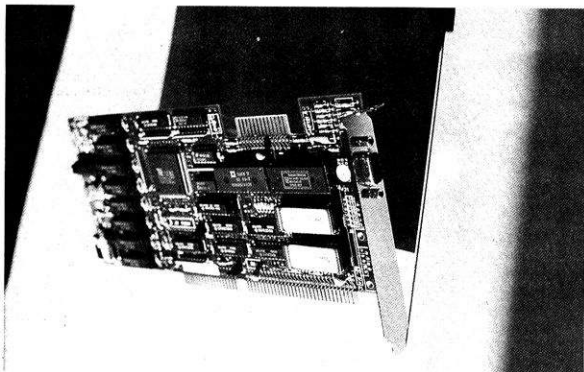
I monitor

Una scheda grafica offre una certa risoluzione ed un ben definito numero di colori, ma solo se collegata ad un monitor dotato di caratteristiche ben precise.

I monitor monocromatici, offerti nei "colori" bianco (oppure ambra o verde) e nero sono disponibili in modelli che presentano diversi ingressi. Quelli con ingresso **videocomposito** sono collegabili a piccoli computers (tra cui il C/64) e sono compatibili solo con la scheda Color Graphic Card. I monitor che offrono un ingresso video **TTL** (**T**ransistor **T**ransistor **L**ogic) sono compatibili con le schede Hercules ed altre monocromatiche digitali.

Più moderni sono i monitor che, nonostante siano monocromatici, sono collegabili a schede VGA che presentano un'uscita a colori **analogica**. Naturalmente non è possibile visualizzare i colori, ma solo varie tonalità di grigio. Questi monitors rappresentano un valido compromesso tra l'esigenza di usare una VGA e la necessità di non spendere cifre elevate.

I monitor a colori con ingresso **RGB** (**R**ed, **G**reen, **B**lu) visualizzano i vari colori grazie alla manipolazione di tre "cannoni" (rosso, verde e blu) controllati dal computer.

Spesso presentano una presa **Scart** dal momento che possono esser collegati anche a videoregistratori e telecamere. Sono compatibili con schede grafiche che presentano un'uscita di questo tipo, come la CGA.

Il meglio dell'attuale produzione è, però, rappresentato dai monitors di tipo **multisync**, che adattano automaticamente la propria frequenza a quella emessa dalla scheda grafica installata sul computer. Sono in grado di visualizzare con precisione fino a 1024 x 768 pixel e dispongono, di solito, anche di connettori per il collegamento a schede grafiche di più modesta tecnologia. I colori visualizzabili sono infiniti e, se il monitor multisync è del tipo monocromatico, i colori vengono simulati da un'infinita varietà di grigi (detta scala di grigi, vale a dire "colori" compresi tra il bianco ed il nero).

Di non trascurabile importanza è la dimensione del cosiddetto **pitch** che, misurato in millimetri, rappresenta la dimensione più piccola del pixel colorabile. Più piccolo è il pitch, migliore è la leggibilità dei caratteri, e dello schermo in generale.

Il valore di **0.28** è indice di un buon monitor; un pitch di **0.39**, di solito, è la caratteristica di monitors più modesti.

Oltre ai monitor di formato standard sono disponibili modelli verticali, che presentano il lato più lungo del video non in posizione orizzontale, come un televisore, ma in posizione verticale. Quasi sempre monocromatici, consentono di lavorare in modo ottimale con programmi di Desk Top Publishing, dal momento che riproducono prefettamente, su video, un foglio di carta di formato A-4.

Richiedono, per il loro funzionamento, una scheda di standard particolare (di solito venduta insieme al monitor stesso) la cui indispensabile installazione sul computer, però, impedisce l'uso di altre schede grafiche. Da usare solo in casi particolarissimi.

Attenti ai programmi

Come avranno notato i nostri lettori, molti programmi commercializzati (perfino i videogames) chiedono che l'utente indichi il tipo di scheda grafica installata.

A mano a mano che il tempo passa, il menu di scelta dei vari pacchetti diventa sempre più ingombrante, paragonabile quasi a quello delle stampanti (se richieste dal programma).

Il motivo è proprio dovuto al continuo progresso tecnologico, ma soprattutto commerciale, di certe schede. Quasi tutti i programmi, oggi, consentono di selezionare la scheda VGA; pochi sono quelli che, al contrario, si "spingono" fino alla SuperVga.

Chi dispone di questa scheda, quindi, si vede costretto ad ammirare, nella massima risoluzione consentita, solo le immagini dimostrative contenute nei floppy della confezione!

Vale, dunque, la pena spender denaro in più per procurarsi una scheda che, a conti fatti, siamo costretti a sfruttare solo in modo parziale?

Il dilemma è d'obbligo; ma se diamo uno sguardo al mercato, ci accorgiamo che tutti i rivenditori offrono super schede, e tantissimi sono quelli che propongono monitor multisync, idonei a sfruttare fino in fondo le caratteristiche delle moderne schede grafiche. Tale particolare non può certo sfuggire ai produttori di software che, nell'aggiornare i loro pacchetti, tenderanno a rendere disponibili risoluzioni grafiche sempre più evolute, che verrebbero apprezzate (ed acquistate) dagli utenti ormai stanchi di vedere solo le immagini demo.

Sarà sufficiente che una sola s/w house proponga un pacchetto grafico utilizzante in modo dignitoso le straordinarie caratteristiche della SuperVga, nello splendore dei suoi 1024 x 768 pixel; tutte le altre s/w house saranno costrette ad imitarla, per non lasciarsi sfuggire le opportunità del mercato.

E in quel momento noi saremo lì, con il nostro bel monitor acceso, pronti a digitare il comando di caricamento del programma e per nulla pentiti di aver speso qualche soldo in più, rinunciando ad un illusorio risparmio.

I computer

Non tutti i computer offrono la possibilità di sostituire la scheda grafica. Alcuni modelli, infatti, montano la circuiteria elettronica, destinata a gestire il video, direttamente sulla scheda madre.

Anche se si dispone di connettori liberi, quindi, non è possibile installare una nuova scheda grafica; questa può entrare in conflitto con quella residente e provocare anche seri guasti.

Alcune schede, inoltre, presentano un connettore del tipo **AT-Bus** (con due pettini di connessioni separati) che ne impedisce l'inserimento nei computer dotati di standard **XT**.

Altri esemplari, poi, pur offrendo la possibilità (teorica) di ospitare altre schede, presentano un numero talmente ridotto di slot che la casa costruttrice è costretta a fornirlo di una scheda grafica che ospita anche la porta seriale e quella parallela.

Sostituendo questa scheda, quindi, dovrete procurarvi una scheda grafica che offra le stesse connessioni, oppure comprare, a parte, una scheda seriale / parallela.

Pertanto la conclusione è unica: prima di acquistare una nuova scheda grafica aprite il computer e verificate i particolari cui abbiamo accennato: eviterete sorprese sgradevolissime.

di Luca Cassioli

STAMPA IN QUATTRO TONI DI GRIGIO

Un programma brevissimo (ovviamente in l.m.) per trasformare, in bianco e nero, le schermate in multicolor del vecchio Commodore 64

Nei numeri scorsi della rivista sono state descritte tecniche valide per "estrarre" schermate multicolori da un qualsiasi programma, salvarle su disco, e poi ricaricarle per manipolarle a piacere.

Con questo programma è ora possibile effettuare la stampa della schermata su carta, cosa finora impossibile, data la differente gestione delle schermate **monocolor** e **multicolor** da parte del C/64.

A questo punto, prima passare alla descrizione del programma e alla spiegazione del suo funzionamento, occorre aprire una parentesi per ricordare quali sono, appunto, le notevoli differenze di cui abbiamo parlato.

Gestione della grafica

Il C/64 dispone, come noto, di due differenti modalità grafiche, oltre al modo testo: il modo **bitmap** standard e il modo **multicolor**.

Il primo tipo di grafica risulta relativamente semplice, in quanto fa uso della notazione binaria per rappresentare punti **accesi** o **spenti** sul video: ogni **byte** della pagina grafica contiene **8 bit**, dei quali quelli settati, cioè posti a 1, sono visibili sullo schermo sotto forma di punti (detti pixels), mentre quelli spenti, naturalmente, no.

La visualizzazione di immagini multicolori risulta invece più complessa: per ogni punto presente sullo schermo saranno infatti occupati due bit dello stesso byte, ed è per questo che in modalità multicolor si dispone di uno schermo di risoluzione minore (**160 * 200** pixel contro i **320 * 200** del modo bitmap standard).

Per colorare, infatti, ogni singolo punto di un colore diverso da quelli circostanti, occorre dapprima memorizzare (da qualche parte, in Ram) la sorgente colore di ciascun punto. In seguito il computer si regolerà agendo, appunto, su quella mappa colore.

L'informazione richiesta viene trascritta nei due bit corrispondenti al punto in questione, che potranno contenere quattro combinazioni i valori binari: **00**, **01**, **10** oppure **11**; cioè, in decimale, **0**, **1**, **2**, oppure **3**. Il valore 0 indicherà che il punto dovrà essere disegnato con il colore dello **sfondo**, il valore 1 indicherà il colore di **primo piano** e i valori 2 e 3 indicheranno, rispettivamente, il colore **multicolor 2** ed il **multicolor 3**.

A questo punto possiamo passare alla descrizione del programma.

Il programma

Il programma presentato, scritto in linguaggio macchina per ovvi motivi di velocità, si occupa del **riconoscimento**

Tabella n. 1 per OR			
Colore 0		**Colore 1**	
decim.	binario	decim	binario
0	0000 0000	64	0100 0000
0	0000 0000	16	0001 0000
0	0000 0000	4	0000 0100
0	0000 0000	1	0000 0001

Tabella n. 2 per OR			
Colore 2		**Colore 3**	
decim.	binario	decim	binario
128	1000 0000	192	1100 0000
32	0010 0000	48	0011 0000
8	0000 1000	12	0000 1100
2	0000 0010	3	0000 0011

della sorgente colore di ciascun punto della pagina grafica. A seconda del parametro inserito dopo la Sys, verranno cancellati tutti i punti che hanno sorgente colore diversa dal valore del parametro. Ad esempio, con...

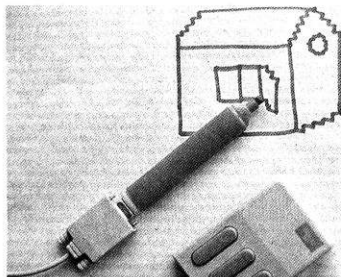
SYS 49152, 0

...verranno privilegiati solo i punti disegnati con il colore di sfondo. Il programma, però, non si limita a cancellare i punti indesiderati, ma provvede anche a porre a 3 (binario = 11) le coppie di bit che interessano. Questo perchè, quando viene effettuata una **hard copy** della pagina grafica, vengono stampati i bytes così come si vedrebbero in modalità grafica standard, dove, come abbiamo visto, esistono solo bit accesi o spenti.

Ponendo le coppie di bit (prima poste a 0) a 1, si otterrà sullo schermo (monocolor) la parte del disegno che (in modo multicolor) era colorata con il colore di sfondo.

Facciamo un esempio:

il byte **00 10 10 11** verrà visto in due modi diversi, a seconda del modo grafico impostato: in modo standard, ad ogni bit acceso corrisponderà un punto. In modo multicolor, invece, si avrà un punto spento (coppia = 00), due punti di colore 2 (coppie = 01), ed un punto di colore 3 (coppia = 11). Dopo il comando SYS 49152, 0, in pagina grafica monocolor si avrà **un solo punto**, "largo" due pixel, **acceso**, e gli altri tre spenti.

Il programma riconosce i 4 diversi tipi di coppie eseguendo prima un **And** tra il byte letto e il valore di una delle 4 tabelle per l'And riportate in figura (una per ogni colore); poi un **Or** con il valore di una delle 4 tabelle per l'OR.

Per il byte **00011011**, ad esempio, che contiene i 4 valori possibili, il programma procede in questo modo:

per parametro = 0:
0001 1011 **AND**
0011 1111 (primo val. pr. tab. And)
0001 1011 (risultato)
0001 1011 **OR**
0000 0000 (primo val. pr. tab. Or)
0001 1011 (risultato)

Il risultato viene confrontato col byte iniziale: se risulterà uguale, vorrà dire che la coppia considerata sarà uguale al parametro, in questo caso 0, e viene quindi posta a 11 (binario), perchè il punto possa essere visualizzato.

In caso contrario, la coppia viene posta a 00, e si passerà alla coppia successiva, per la quale saranno considerati i secondi valori delle tabelle, e così via per la terza e la quarta coppia.

Alla fine del procedimento, il risultato, nel nostro caso, sarà **11 00 00 00**, cioè dei 4 punti verrà stampato solo 1 (ed esattamente il primo, che era spento), e quindi colorato con il colore di sfondo. Lo stesso procedimento viene eseguito per gli altri valori del parametro.

Da notare, però, che la stampa del colore di fondo sarà realmente necessaria solo nel caso in cui, nel disegno, lo sfondo sarà colorato di nero o di un colore scuro, la cui presenza, sulla carta, sia indispensabile per la comprensibilità del disegno.

Un'ultima nota sul programma: questo visualizza l'immagine da stampare nella bitmap allocata da **$2000** (dec. **8192**) a **$3f3f** (decimale **16191**), mentre legge i bytes da elaborare a partire da **$6000** (dec. **24576**), cioè dalla zona in cui vengono memorizzate le schermate **Koala**. Questo per rendere più facile la stampa delle stesse, in quanto sarà sufficiente caricare il file contenente i dati della

Colore 0		**Colore 1**	
decim.	binario	decim.	binario
63	0011 1111	127	0111 1111
207	1100 1111	223	1101 1111
243	1111 0011	247	1111 0111
252	1111 1100	253	1111 1101

Tabella n. 1 per AND

Colore 2		**Colore 3**	
decim.	binario	decim.	binario
191	1011 1111	255	1111 1111
239	1110 1111	255	1111 1111
251	1111 1011	255	1111 1111
254	1111 1110	255	1111 1111

Tabella n.2 per AND

schermata (utilizzando il suffisso **,8,1**), e poi utilizzare il programma in questione.

Come usare il programma

Anche se il **meccanismo** del programma può sembrare eccessivamente complesso ai meno esperti, l'**utilizzo** dello stesso è molto semplice.

Dopo aver digitato (e salvato!) il caricatore Basic, prestando particolare attenzione alle righe Data, digitate Run. Nel caso di errori di digitazione, il programma si arresterà per consentirne il controllo.

In caso contrario apparirà una riga di spiegazione:

Attivare con SYS 49152, x

Prima di procedere all'attivazione, occorre caricare in memoria una routine di hard copy della pagina grafica, scegliendola in modo che non interferisca con la routine di queste pagine.

Prima di attivarla, tenete presente che il valore **X** deve essere compreso tra **0** e **3** (con le corrispondenze viste prima) e che il programma non controlla la correttezza di tale valore, per cui inserendone altri si otterranno risultati indesiderati.

Una volta avvenuta la conversione (confermata dalla comparsa del consueto Ready) si potrà procedere con la stampa, per la quale si dovranno prendere opportuni provvedimenti.

Volendo, si può osservare il funzionamento del programma attivando la pagina grafica con...

Poke 53272, 29

...e...

Poke 53265, 59

In questo caso, però, si dovranno impartire i comandi alla cieca, dal momento che al posto dei caratteri e del cursore compariranno quadratini colorati privi di senso.

Dopo aver inserito il foglio nella posizione voluta, effettuate un segno a matita in modo che, al termine della stampa, il foglio stesso possa essere riportato nella posizione primitiva con la maggior precisione possibile, pena l'illeggibilità del disegno.

Prima di procedere con la stampa, bisognerà decidere che colore dovrà avere la parte del disegno sul foglio. Il programma, infatti, serve per stampare i disegni in **quattro toni di grigio**, cioè in 4 colori, e bisognerà stabilire a priori il tono di grigio corrispondente alla parte di disegno da stampare.

In generale, per stampare schermate in cui è necessaria la stampa dello sfondo in nero (per la leggibilità del disegno), si dovrà procedere in questo modo, dopo aver caricato la schermata in memoria:

1) digitare Sys 49152, 0
2) posizionare il foglio
3) stampare (prima volta)
4) riposizionare foglio
5) stampare (seconda volta)
6) riposizionare foglio
7) stampare (terza volta)
8) digitare: Sys 49152, 1
9) stampare due volte riposizionando il foglio ogni volta
10) digitare: SYS 49152, 2
11) stampare una volta dopo aver riposizionato il foglio.

In questo modo si otterrà un disegno in 4 colori, di cui uno, il colore 3, sarà il "**bianco**" del foglio. Questo permetterà di risparmiare tempo; per ottenere, però, 4 toni di grigio, bisognerà stampare 4 volte il colore 0; 3 volte il colore 1; 2 volte il colore 2, ed una volta il colore 3 (per un totale di 4 + 3 + 2 + 1 = 10 stampe, contro le 6 di prima).

In entrambi i casi la procedura è un po'(!) lunga, ma il risultato vale l'attesa.

E' importante ricordare, prima di iniziare la stampa, di settare la stampante in modo da **non** "saltare" righe ogni qualvolta **crede** di aver stampato un foglio intero. Propriamente, bisogna porre a 0 il piede del modulo. Questo per evitare di rovinare il disegno in fase di stampa a causa di un allineamento non ottimale.

Un'ultima precisazione: alcune schermate, se divise in 4 colori, risultano non stampabili, non per problemi pratici, ma estetici. Infatti, il disegno apparirà a scale in alcuni punti, come accade colorando un disegno in modo monocolor. Per il resto, non ci sono problemi.

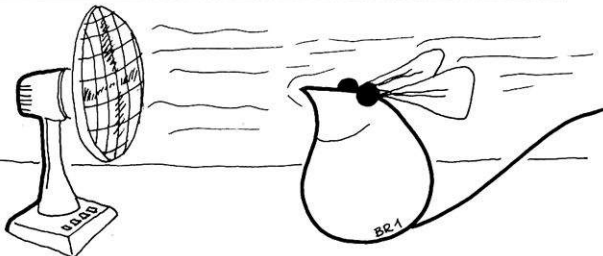
Un esempio pratico

Per avere un'idea dei risultati ottenibili, con il gioco **Ghostbusters II** provate a stampare la schermata che si trova in memoria quando si è giunti all'ultimo quadro. Se pensate di non raggiungere l'ultimo quadro, per la difficoltà del gioco, niente paura: è sufficiente rinominare i vari files della directory in modo

```
10 rem ** stampa immagini multicolor **
20 rem **   by  luca cassioli        **
30 rem **    s.angelo romano  (roma) **
40 rem **  versione per commodore 64 **
50 :
60 fora=49152to49295:readb:pokea,b:c=c+b:next
70 ifc<>18393thenprint"errore nei data":stop
80 print"attivare con:":print"sys49152,x  -  x compreso tra 0 e 3"
100 data032,069,192,234,189,118,192,141,028,192,056,233,004,141,025,192
110 data189,000,096,160,000,141,044,193,057,078,192,025,082,192,205,044
120 data193,208,028,025,110,192,200,192,004,208,234,157,000,064,232,208
130 data223,238,018,192,238,045,192,173,045,192,201,064,208,210,096,057
140 data114,192,076,038,192,076,127,192,234,234,234,234,234,234,063,207
150 data243,252,000,000,000,000,127,223,247,253,064,016,004,001,191,239
160 data251,254,128,032,008,002,255,255,255,255,192,048,012,003,192,048
170 data 12,003,063,207,243,252,082,090,098,106,000,000,000,000,000,169
180 data 96,141,018,192,169,032,141,045,192,032,253,174,032,158,183,096
190 end

ready.
```

```
. 1c000  20 45 c0 jsr $c045    :prepara locazioni.
. 1c003  ea       nop          :***
. 1c004  bd 76 c0 lda $c076,x  :carica valore per leggere tabella appropriata.
. 1c007  8d 1c c0 sta $c01c    :lo salva.
. 1c00a  38       sec          :sottrae 4
. 1c00b  e9 04    sbc #$04     :(una tabella = 4 bytes)
. 1c00d  8d 19 c0 sta $c019    :memorizza nuovo valore.
. 1c010  bd 00 60 lda $6000,x  :carica byte da pagina grafica 2(inizio-$4000).
. 1c013  a0 00    ldy #$00     :azzera contatore.
. 1c015  8d 2c c1 sta $c12c    :salva byte letto.
. 1c018  39 4e c0 and $c04e,y  :esegue and con valore della prima tabella.
. 1c01b  19 52 c0 ora $c052,y  :esegue or con valore della seconda tabella.
. 1c01e  cd 2c c1 cmp $c12c    :se byte elaborato diverso da byte letto,
. 1c021  d0 1c    bne $c03f    :spegne coppia di bit,
. 1c023  19 6e c0 ora $c06e,y  :altrimenti la accende.
. 1c026  c8       iny          :incrementa contatore.
. 1c027  c0 04    cpy #$04     :ripete
. 1c029  d0 ea    bne $c015    :4 volte.
. 1c02b  9d 00 20 sta $2000,x  :salva byte elaborato in pag. grafica 1 ($2000).
. 1c02e  e8       inx          :ripete
. 1c02f  d0 df    bne $c010    :fino
. 1c031  ee 12 c0 inc $c012    :alla
. 1c034  ee 2d c0 inc $c02d    :fine
. 1c037  ad 2d c0 lda $c02d    :della
. 1c03a  c9 40    cmp #$40     :pagina
. 1c03c  d0 d2    bne $c010    :grafica.
. 1c03e  60       rts          :fine.
. 1c03f  39 72 c0 and $c072,y  :azzera coppia di bit
. 1c042  4c 26 c0 jmp $c026    :e continua.
. 1c045  4c 7f c0 jmp $c07f    :salta a subroutine posizionata dopo le tabelle.

         *** tabelle ***

. 1c07f  a9 60    lda #$60     :prepera locazioni
. 1c081  8d 12 c0 sta $c012    :per lettura
. 1c084  a9 20    lda #$20     :e per
. 1c086  8d 2d c0 sta $c02d    :scrittura disegno.
. 1c089  20 fd ae jsr $aefd    :preleva parametro
. 1c08c  20 9e b7 jsr $b79e    :dopo sys
. 1c08f  60       rts          : e torna
```

che il file "**2. First Story**" diventi "**4. Second Story**" ed il file "**3. The Cloaks**" diventi "**5. The Liberty**". Ora caricate il gioco, senza far caso alle escalamazioni che compariranno sullo schermo se avrete commesso qualche errore di ridenominazione... iniziate ora a giocare e, in un qualsiasi momento, premete il tasto Return.

La schermata che compare può essere stampata come la vedete (se possedete un monitor in bianco e nero...) utilizzando la routine pubblicata.

di Fabrizio Bazzo

UNA FUSIONE TRA BASIC E L.M.

Il presente articolo sarà molto apprezzato da coloro che, pur lavorando in Basic, hanno spesso a che fare con procedure che richiamano una routine in linguaggio macchina

Ormai sono moltissimi gli utenti del C/64 che dispongono di estensioni di linguaggio e/o routines l.m. di vario genere per supplire alla scarsa potenza del Basic 2.0; sono un po' meno coloro che ne fanno uso realizzando programmi Basic farciti di comandi non standard e di **Sys** in zone di memoria più o meno "alte".

Il risultato del lavoro di codesti loschi individui consiste quindi, in genere, in un unico file Basic che contiene, tra le prime istruzioni, la verifica della presenza della routine e si incarica, se è il caso, di caricarla in memoria.

I programmatori più ordinati (cioè nessuno) salvano la routine di cui parliamo (che d'ora in poi chiameremo **Rout**) e tutti i programmi che la utilizzano su un unico disco; gli altri (cioè tutti) memorizzano abitualmente, in seguito a sviluppi, modifiche, innesti e copie, i due files Basic e Rout ripartiti casualmente tra le migliaia di dischi posseduti, la maggior parte dei quali, tradizionalmente, senza etichetta.

La soluzione è, ovviamente, unire Basic e Rout in un unico file da caricare, lanciare, e successivamente smarrire senza problemi.

Il programma **Mini Linker** si incarica proprio di questo, ma richiede per il corretto utilizzo un minimo di attenzione dal momento che la procedura non è (nè potrebbe essere) completamente automatica.

Supponiamo di avere i due files, **Basic** e **Rout**, sul medesimo disco, e **Mini Linker** già presente in memoria in forma di **codice macchina** e non in forma di caricatore Basic (da caricare con **,8,1** per intenderci e non così come è pubblicato in queste pagine).

Carichiamo ora il nostro programma **Basic** (che dovrà utilizzare **Rout**) e aggiungiamo la seguente linea:

0 sys 1999

... in cui il numero 1999 (di 4 cifre, preceduto da uno spazio) serve per far "spazio" nel listato Basic e verrà sostituito tra poco. Leggiamo infatti i puntatori di inizio variabili con...

Print Peek (45) + Peek (46) * 256

Il valore ottenuto è, come i più furbi sanno, l'argomento corretto da "passare" alla **Sys**: modifichiamo quindi la linea **0** di conseguenza rispettando, cioè, la spaziatura se il valore è di 4 cifre, annullandola se di 5 cifre; ora lanciamo Mini-linker con:

Sys 49152, "ROUT", Entry, NP

Entry è l'indirizzo di partenza di Rout, e **NP** il numero di pagine di memoria che occupa, ossia la lunghezza in **bytes / 256** e arrotondata per eccesso: corrisponde (con buona approssimazine) al numero dei blocchi occupati sul disco.

Rout verrà caricata nell'area Basic, e quindi verrà salvato un file di nome **Pack** il quale, una volta lanciato, si occuperà di porre Rout nell'area di memoria originale e di rendere innocua la Sys di linea 0, che potrebbe causare problemi in caso di lancio successivo del programma.

Con lo stesso sistema si possono naturalmente linkare **set di caratteri, musiche da interrupt, schermate**, e così via.

Il **Macro Assembler** si è rivelato uno strumento indispensabile in fase di sviluppo, soprattutto per i riferimenti alle locazioni interne del programma, in parte automodificante.

I felici possessori del potente package possono tranquillamente copiare il sorgente; chi ne fosse sprovvisto può usare il caricatore Basic,avendo cura di ripristinare i puntatori con un **New** dopo averlo lanciato, prima di cominciare a *sposare* files.

Dato che non vogliamo offendere nessuno raccontando che cosa sono i puntatori del Basic e che cosa fanno le routine del S.O. (peraltro tra le più gettonate) che compaiono nel listato sorgente, rimandiamo agli esaurienti articoli pubblicati sui numeri **55**, **56** & **61** di C.C.C.

```
10 REM
20 REM    MINI-LINKER
30 REM BY FABRIZIO BAZZO
40 REM
100 DATA 032,253,174,032,158,173,032,130
110 DATA 183,132,002,160,000,177,034,153
120 DATA 000,193,200,196,002,208,246,032
130 DATA 253,174,032,138,173,032,247,183
140 DATA 140,105,192,141,109,192,032,253
150 DATA 174,032,158,183,142,113,192,165
160 DATA 045,141,081,192,166,046,142,082
170 DATA 192,024,105,040,141,097,192,141
180 DATA 146,192,144,001,232,142,101,192
190 DATA 142,148,192,162,000,189,091,192
200 DATA 157,240,240,232,224,040,208,245
210 DATA 076,131,192,169,143,141,005,008
220 DATA 169,255,133,247,169,255,133,248
230 DATA 169,255,133,249,169,255,133,250
240 DATA 162,000,160,000,177,247,145,249
250 DATA 200,208,249,230,248,230,250,202
260 DATA 208,240,096,032,188,192,165,002
270 DATA 162,000,160,193,032,189,255,169
280 DATA 000,162,255,160,255,032,213,255
290 DATA 142,181,192,140,183,192,032,188
300 DATA 192,169,004,162,198,160,192,032
310 DATA 189,255,169,008,162,001,134,247
320 DATA 133,248,169,247,162,255,160,255
330 DATA 032,216,255,096,169,015,162,008
340 DATA 160,000,032,186,255,096,080,065
350 DATA 067,075
360 :
370 CK=0:CT=29917
380 FORA=49152TOA+201:READB:POKEA,B
385 CK=CK+B:NEXT
390 IFCK<>CTTHENPRINT"DATA ERROR":STOP
400 END

READY.
```

```
;-----------------------------
; mini basic-lm linker
;  by fabrizio bazzo
;-----------------------------
*     =$c000
nomel=$c100
;-----------------------------
getnam jsr $aefd    ;prende il nome
       jsr $ad9e    ;della routine
       jsr $b782    ;e la
       sty $02      ;lunghezza
       ldy #$00     ;e li pone
scan   lda ($22),y ;in $c100
       sta nomel,y ;e $02
       iny
       cpy $02
       bne scan
;-----------------------------
input  jsr $aefd    ;prende l'entry
       jsr $ad8a    ;della routine
       jsr $b7f7    ;e setta i pa-
       sty init+9   ;rametri (to)
       sta init+13 ;di init
       jsr $aefd    ;prende il n.
       jsr $b79e    ;di pagine
       stx transf+1 ;da trasferire
;-----------------------------
       lda $2d      ;legge la fine
       sta ttr+6    ;del basic e
       ldx $2e      ;prepara il
       stx ttr+7    ;trasferimento
       clc
       adc #$28     ;lenght
       sta init+1
       sta load+15 ;setta (from)
       bcc cont     ;per init e
       inx          ;prepara e re-
cont   stx init+5   ;gistri per
       stx load+17 ;il load
;-----------------------------
ttr    ldx #$00     ;pone token
cont1  lda token,x ;e transf
       sta $f0f0,x ;a ridosso
       inx          ;del basic
       cpx #$28
       bne cont1
;-----------------------------
       jmp load     ;load routine
;-----------------------------
token  lda #$8f     ;sys --> rem
       sta $0805
;-----------------------------
init   lda #$ff     ;routine:
       sta $f7      ;from ($f7)
       lda #$ff
       sta $f8
       lda #$ff     ;to ($f9)
       sta $f9
       lda #$ff
       sta $fa
;-----------------------------
transf ldx #$00     ;lavori
cont3  ldy #$00
cont2  lda ($f7),y
       sta ($f9),y
       iny
       bne cont2    ;in
       inc $f8
       inc $fa
       dex
       bne cont3    ;corso
       rts
;-----------------------------
load   jsr setfil   ;carica la
       lda $02      ;routine a
       ldx #<nomel ;ridosso
       ldy #>nomel ;di transf
       jsr $ffbd
       lda #$00
       ldx #$ff
       ldy #$ff
       jsr $ffd5
;-----------------------------
       stx save+23 ;ultima loc.
       sty save+25 ;occupata
;-----------------------------
save   jsr setfil
       lda #$04     ;setta alcuni
       ldx #<nomes ;parametri
       ldy #>nomes ;per il
       jsr $ffbd    ;ram save
       lda #$08
       ldx #$01
       stx $f7
       sta $f8
       lda #$f7
       ldx #$ff
       ldy #$ff
       jsr $ffd8    ;e salva
       rts
;-----------------------------
setfil lda #$0f     ;file
       ldx #$08     ;device
       ldy #$00     ;2.nd address
       jsr $ffba
       rts
;-----------------------------
nomes .byte 'pack'
.end
```

di Giancarlo Mariani

UN FILE SERVITORE DI TRE PADRONI

Il colloquio con il drive viene eseguito in modo pressochè identico usando i tre linguaggi più noti: QuickBasic, Turbo- C e Turbo-Pascal

Il presente articolo, pur continuando la serie dedicata al **linguaggio C**, offre la novità di presentare una procedura, unica, implementata in più linguaggi.

Ribadiamo ancora una volta l'importanza nel conoscere più linguaggi invece di uno solo, sia perchè un linguaggio può risultare più adatto di un altro per risolvere un determinato problema, sia per non fossilizzarsi su tecniche di cui si è già padroni.

La conoscenza di un linguaggio di programmazione, inoltre, facilita l'apprendimento degli altri, soprattutto perchè molto spesso **le differenze tra un linguaggio ed un altro non sono così marcate come (erroneamente) si ritiene.**

Il listato di queste pagine, per la breve lunghezza e la notevole semplicità, può essere compreso dai principianti senza esser costretti a digitarlo, e consentirà un raffronto diretto tra i tre linguaggi scelti per presentare il **tema** di questo mese (**gestione di un file sequenziale**).

Grazie all'estrema semplicità della procedura, è probabile che il listato proposto giri anche su versioni più vecchie di quelle da noi usate per svilupparla (**QuickBasic 4.5** della Microsoft, il **Turbo-Pascal 5.5** ed il **Turbo-C 2.0**, entrambi della Borland).

Il problema

Supponiamo che, in un negozio di alimentari, sia visibile la lista dei prezzi degli articoli in vendita. Osservando il cartello, e sapendo quali articoli acquistare, potremo fare un rapido calcolo della somma da spendere. Ammettiamo, ad esempio, che il cartellone indichi...

Pasta, 2500 (Kg)
Riso, 4000 (Kg)
Insalata, 1500 (Kg)
Focaccia, 500 (1)
Pane, 3500 (Kg)

Per comprare 2 Kg di pasta, 1Kg di riso, 3 focacce e 1/2 Kg di pane, dovremo quindi spendere:

2 x 2500 + 1 x 4000 + 3 x 500 + 1/2 x 3500 = 12250

I programmi proposti eseguono quanto detto. L'archivio dal quale vengono "estratte" le merci, ed i prezzi corrispondenti, è un semplicissimo **file** il cui formato è il seguente:

Articolo1, Prezzo1<CR>
Articolo2, Prezzo2<CR>
...
ArticoloN, PrezzoN<CR>

In pratica, nel file (che può essere addirittura creato con qualsiasi editor o word-processor in grado di generare un file in codice Ascii) andrà inserito:
- il nome dell'articolo,
- il carattere virgola (,),
- il prezzo (sempre in codici ASCII),
- il carattere Return (Ascii 13).

Ogni riga del file sarà quindi composta da un articolo affiancato dal relativo prezzo. Nel caso della lista precedente, il file risulterebbe...

Pasta,2500<CR>
Riso,4000<CR>
Insalata,1500<CR>
Focaccia,500<CR>
Pane,3500<CR>

Naturalmente il file deve essere precedentemente creato con il nome di MAGAZ.TXT *PRIMA* di mandare in

```
/* Programma Ricerca - TurboC */
```

L'istruzione **include** permette, appunto, di includere altri files nel testo sorgente. Questi sono normalmente definizioni di prototipi standard del C che contengono la parte della dichiarazione delle procedure e funzioni del linguaggio C. L'omissione di questi file può produrre segnalazione di **Warnings** in fase di compilazione.

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>
```

Queste istruzioni sono la copia di quelle usate per dichiarare l'inizio di procedure e funzioni; oltre a questo compito, permettono al compilatore di controllare la chiamata ed il passaggio dei parametri alle stesse subroutines.

```
/* Prototipi di funzioni */
int Ricerca (char *a, long *Prezzo);
void VediLista (void);
```

E' la dichiarazione delle variabili, obbligatoria in C, pena segnalazione di errore. Le variabili sono **globali**, ossia visibili dall'intero programma, comprese procedure e funzioni.

```
long Totale, Prezzo, Temp;
char a[80];
int q;
```

Rappresenta l'inizio della **Funzione Ricerca** (di tipo intero), che ha due parametri, uno **stringa** ed uno di tipo **Long**, ma comunque entrambi di tipo puntatore (*) dato che devono venir modificati dalla funzione stessa. Questa deve essere racchiusa tra la parentesi graffa aperta ({) e chiusa (}). Nella funzione possono essere dichiarate variabili come locali, allo scopo di non esser "viste" in nessun'altra sezione.

```
/* Funzione ricerca */
int Ricerca (char *a, long *Prezzo)
{
int Trovato;
FILE *stream;
char Temp[80],Temp2[80];

Trovato = 0;
stream = fopen ("magaz.txt","rt");
```

L'istruzione **While** apre un ciclo, che sarà abbandonato quando la condizione specificata tra parentesi risulterà **falsa**. Le istruzioni appartenenti al ciclo vanno racchiuse tra parentesi graffe. L'istruzione **feof** diventa vera quando viene raggiunta la fine del file indicato. Il punto esclamativo (!) posto prima della funzione serve per negarla. In pratica, questa riga è l'equivalente della riga basic **While Not Eof (1)**.

```
while (!feof(stream))
{
fscanf(stream,"%s",Temp); Temp[strlen(Temp)-
1]=0;
fscanf(stream,"%s",Temp2);
```

L'istruzione **strupr** trasforma la stringa specificata in caratteri maiuscoli.

```
strupr(Temp); strupr(a);
if (strcmp(Temp,a)==0) { Trovato=1; break; }
}
fclose (stream);
*Prezzo = atol(Temp2);
```

Tramite l'istruzione **return** si fa assumere alla funzione Ricerca il valore specificato, in modo da farlo controllare dal main (vedi if Ricerca...).

```
return (Trovato);
}
```

Inizia la **Procedura Vedilista**. Le istruzioni appartenenti alla procedura devono essere racchiuse tra parentesi graffe; anche qui possono essere dichiarate variabili locali.

```
/* Procedura vedilista */
void VediLista (void)
{
FILE *stream;
char Temp[80];

stream = fopen ("magaz.txt","rt"); printf("\n");
while (fgets(Temp,80,stream)!=NULL) printf ("Ar-
ticolo, prezzo : %s",Temp);
fclose (stream); printf("\n");
}
```

L'istruzione **main** segna l'inizio del main program, o programma principale, anch'esso racchiuso tra graffe.

```
/* main program */
void main ()
{
Totale = 0;
clrscr();
```

L'istruzione do apre un ciclo, da chiudere con l'istruzione while. Anche do può essere eseguita da while, ma in questo caso non è necessaria la chiusura. Le istruzioni appartenenti al ciclo devon essere racchiuse tra graffe. L'istruzione break permette di uscire dal ciclo in qualsiasi momento, anche se non si sono verificate le condizioni necessarie.

```
do
```

```
{
a[0]=0; printf("\nInserisci l'articolo (* per termina-
re) : ");
scanf("%s",a);
if (a[0]=='*') break;
```

L'istruzione **if,** strutturata, consente forme del tipo **if.... else**. In questo caso, **if Ricerca** richiama la **funzione** Ricerca, e si comporta a seconda del risultato fornito da questa.

```
if (Ricerca(a, &Prezzo)==0)
{
printf("\nArticolo non trovato. vuoi vedere la lista
(S/N) : ");
scanf ("%s",a);
```

Per richiamare le procedure non è necessaria **Call** come in Qbasic, ma è sufficiente specificarne il nome, seguito dagli eventuali parametri posti tra parentesi. In questo caso viene richiamata **VediLista**. Le parentesi sono obbligatorie, anche se non vi è alcun parametro da passare.

```
if (a[0]=='s' || a[0]=='S') VediLista();
}
```

L'istruzione **else** è l'alternativa a **if**; ossia, se la condizione specificata da if è vera, vengono eseguite tutte le istruzioni poste, tra parentesi graffe, dopo if altrimenti vengono eseguite le istruzioni, sempre poste tra graffe, dopo else.

```
else
{
printf("\nArticolo : %s , Prezzo : %ld, Quantita :
",a,Prezzo);
scanf ("%d",&q); if (q<0) q=0;
printf ("\nPrezzo totale : %ld\n",q*Prezzo);
Totale = Totale + q * Prezzo;
}
}
while (a[0]!=0);
printf("\nSpesa totale: %ld\n", Totale);
}
```

I linguaggi

I tre linguaggi **Turbo-Pascal**, **Turbo-C** e **QuickBasic** sono tra gli ultimi sviluppati per sistemi **Ms-Dos**, e rappresentano potentissimi strumenti non solo di studio, ma anche di lavoro per applicazioni professionali.

Il QuickBasic della **Microsoft**, pur riuscendo a far girare perfettamente e senza nessuna modifica i programmi scritti in **GWbasic**, non è una semplice derivazione di questo, ma rappresenta una vera e propria rivoluzione nel campo della programmazione in Basic: I numeri di linea sono spariti (anche se opzionalmente è ancora possibile utilizzarli, grazie alla compatibilità con il GW-Basic); è possibile la programmazione strutturata (quella dei linguaggi tipo Pascal, per intenderci) con cicli **Do ... Loop While / Until, If .. Then .. Else .. EndIf, Select Case**, ed altri; permette la dichiarazione delle variabili e delle subroutines usate nel programma; consente di gestire (come variabili locali all'interno di procedure e funzioni) variabili che hanno lo stesso nome di quelle globali.

Inoltre le **procedure** e le **funzioni**, alla pari di altri linguaggi, sono separate dal **Main Program** (programma principale), e possono trasferire parametri o leggere valori di ritorno. E' anche possibile la ricorsività (impossibile in GW-Basic).

La **velocità** di esecuzione di un programma scritto in QuickBasic è di circa 5 volte superiore a quella dello stesso listato, ma scritto in GW-basic; è anche possibile creare direttamente un file eseguibile, che non ha bisogno del compilatore per girare.

Il **Turbo-C** ed il **Turbo-Pascal** della **Borland** sono due tra i più potenti compilatori presenti sul mercato Ms-Dos.

Essi, alla pari del QuickBasic, possiedono un **ambiente integrato** comprendente **Editor**, **Compilatore**, **Debugger**, **Help** in linea, ed altre utility o facilitazioni.

In pratica, è possibile passare direttamente dalla scrittura del testo sorgente alla compilazione (ed alla successiva esecuzione) semplicemente premendo un tasto. Inoltre il debugger integrato permette di osservare il **flusso** logico del programma per individuare eventuali errori di programmazione.

La velocità di compilazione è spaventosa e, **specialmente nel caso del Turbo-C**, l'esecuzione dei programmi compilati è rapida quasi quanto procedure scritte direttamente in **Assembly**.

Non ultima, merita particolare menzione il **prezzo** degli stessi linguaggi, abbordabilissimo non solo da professionisti, ma anche da **studenti** che vogliono dilettarsi nella programmazione. La documentazione, inoltre, è molto ben curata, sia nelle versioni originali inglesi, sia in quelle italiane.

```
DEFINT A-Z
REM Programma Ricerca - QuickBasic
```

Le righe Declare... potrebbero anche essere non digitate, dato che le aggiunge automaticamente il compilatore all'atto del salvataggio del programma; servono per dichiarare eventuali procedure e funzioni annesse al programma per permettere al compilatore di controllare la chiamata ed il passaggio dei parametri alle subroutines.

```
DECLARE FUNCTION Ricerca% (a$, Prezzo AS LONG)
DECLARE SUB VediLista ()
```

Le istruzioni Dim... dichiarano le variabili usate dal programma. In QuickBasic non è necessario (ma è comunque permesso) dichiararle. Le variabili **Totale**, **Prezzo** e **Temp** sono, in questo caso, da dichiarare, o meglio, da dimensionare obbligatoriamente, dal momento che, a differenza delle altre (che sono intere) queste sono di tipo **Long**, ossia intero doppio.

```
DIM Totale AS LONG
DIM Prezzo AS LONG
DIM Temp AS LONG
CLS : PRINT "Attenzione al Path ove posizionare"
PRINT "file di nome MAGAZ.TXT": PRINT
Totale = 0
```

L'istruzione **Do** apre un ciclo, che può essere chiuso con le istruzioni **Loop While** oppure **Loop Until**. Anche Do può essere eseguita da While o Until. Se entrambe le istruzioni Do e Loop sono inserite senza While o Until (come in questo caso) il ciclo risulta **infinito**, ed il solo modo per uscirne è l'istruzione **End**, che termina il programma, oppure **Exit Do**, che salta all'istruzione successiva a Loop.

```
DO
a$ = "": PRINT
INPUT "Inserisci l'articolo (* per terminare)"; a$
IF a$ = "*" THEN EXIT DO
```

L'istruzione **If** può essere strutturata, consentendo forme tipo **If.... Then.... Else.... EndIf**. In questo caso, **If Ricerca** richiama la **funzione** Ricerca, e quindi si comporta a seconda del risultato fornito da questa.

```
IF Ricerca(a$, Prezzo) = 0 THEN
PRINT : INPUT "Articolo non trovato. vuoi vedere la lista (S/N)"; a$
```

L'istruzione **Ucase** trasforma i caratteri della stringa fornita in maiuscolo. **Call** richiama la procedura specificata.

```
IF UCASE$(a$) = "S" THEN CALL VediLista
ELSE
```

Else è l'alternativa ad If: se la condizione specificata da If è vera, vengono eseguite le istruzioni poste tra **If...Then** ed **Else**, altrimenti vengono eseguite quelle tra **Else** ed **End If**.

```
PRINT : PRINT "Articolo :"; a$; ", Prezzo :"; Prezzo; ", ";
INPUT "Quantita' :"; q: IF q < 0 THEN q = 0
PRINT "Prezzo totale : "; q * Prezzo
Totale = Totale + q * Prezzo
END IF
LOOP
PRINT : PRINT "Spesa totale: "; Totale
END
```

Questo è l'inizio della **funzione Ricerca**, che ha due parametri, il primo di tipo stringa ed il secondo di tipo Long. La chiusura della funzione viene effettuata tramite End Function.

```
FUNCTION Ricerca (a$, Prezzo AS LONG)
Trovato = 0
OPEN "c:\mariani\file\magaz.txt" FOR INPUT AS #1
```

La funzione **Eof(file)** diventa vera quando il file specificato è giunto alla fine. Quindi, il ciclo **Do While Not Eof (1)...Loop** si ripeterà sino alla fine del file magaz.txt.

```
DO WHILE NOT EOF(1)
INPUT #1, b$, c$
IF UCASE$(a$) = UCASE$(b$) THEN Trovato = 1: EXIT DO
LOOP
CLOSE #1
Prezzo = VAL(c$)
Ricerca = Trovato
END FUNCTION
```

L'istruzione di assegnazione che ha come nome della variabile il nome della funzione, produce il risultato di assegnare a quella funzione il valore specificato, in modo che possa essere controllato dalla If Ricerca nel main.

```
SUB VediLista
CLS
OPEN "c:\mariani\file\magaz.txt" FOR INPUT AS #1
DO WHILE NOT EOF(1)
INPUT #1, b$, c$
IF VAL(c$) > 0 THEN
a$ = LEFT$("Articolo : " + b$ + SPACE$(40), 40)
PRINT a$; "Prezzo : "; c$
END IF
LOOP
CLOSE #1
END SUB
```

Un mondo pieno di caverne vi attende per mettere alla prova la vostra abilità

Computer: Amiga inespanso
Gestione: Joysytick
Tipo: Arcade a caverne
Softhouse: Electronic Arts

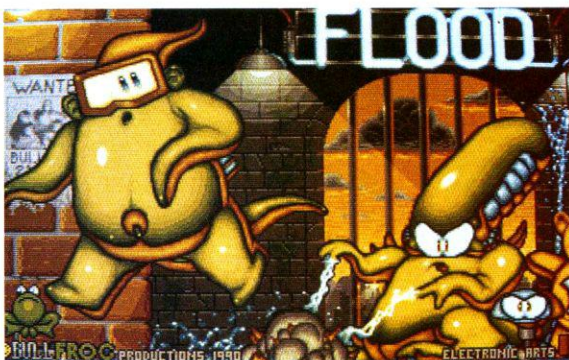
FLOOD

La **Bullfrog**, dopo i leggendari **Populous** e **Fusion**, ha affidato alla **ECA** il proprio nuovo programma, curatissimo nell'aspetto grafico.

Il gioco

Il protagonista di Flood è un piccolo simpaticone chiamato **Quiffy**. E' l'ultimo sopravvissuto della razza dei Blobbie, sterminati dai perfidi Psycho Teddies. Quiffy deve perciò muoversi per una marea di caverne, su circa 40 piani, affrontando pericoli di tutti i tipi, come trappole, trabocchetti e curiosi, pericolosissimi mostriciattoli.

L'uscita di ogni caverna è un teletrasportatore che può essere utilizzato solo quando Quiffy ha raccolto tutti gli oggetti sparsi per la caverna.

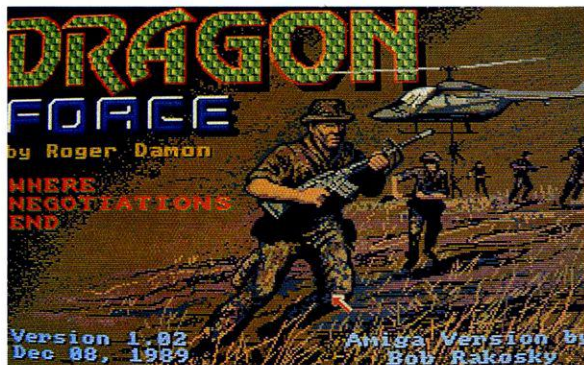
Non mancano, come in ogni gioco a caverne che si rispetti, le armi: granate, dinamite e lanciafiamme, ad esempio, molto utili per instillare un po' di ragione ai mostri nemici.

La tecnica

Secondo lo stile Bullfrog, la grafica è estremamente curata: tutto è molto ricco di particolari e colorato, pur nella scarsa luce delle caverne! Gli sprites hanno chiari connotati da "cartone animati", sebbene l'uso dei colori e delle ombreggiature fornisca una apparenza molto solida e realistica, non comune nei giochi di questo tipo, sólitamente votati soltanto alla velocità ed al numero di oggetti e mostri.

Anche gli effetti sonori sono validi, giostrando una marea di effetti di urletti, passi, gocciolii ed altri suoni curiosissimi.

Il voto

Un caposcuola nel suo genere per raffinatezza grafica. Ma poco è l'intelletto richiesto. 7+.

DRAGON FORCE

Un gioco di strategia, da giocare con pazienza contro il computer

Computer: Amiga inespanso
Gestione: Joystick
Tipo: Arcade strategico
Softhouse: Interstel

Il Drastic Response Assault Group Operation Network è in città, e questo significa "affari": un gruppo di 14 mercenari specializzati nella lotta contro il terrorismo ai nostri ordini. Programma vagamente simile a **Laser Squad**.

Il Gioco

Si tratta di un gioco da guerra per una sola persona che affronta le mosse del computer. Sono previste alcune dozzine di missioni, affrontabili in qualunque ordine, secondo la scelta fatta dal menu iniziale. Una volta scelta la missione, ed i combattenti che ci aiuteranno, è necessario, come al solito in questo tipo di giochi, scegliere anche le armi da usare. Dal momento che ogni soldato ha dei connotati fisiologici e bellici, è necessario scegliere le armi opportune per ciascuno di essi.

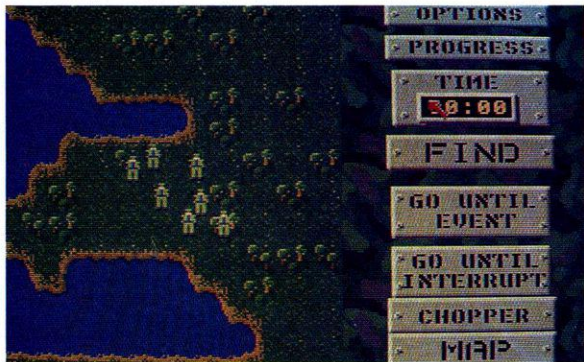
Il gioco si svolge in **rounds**: si possono dare sino a cinque ordini (brevi e molto specifici) dalla consolle a ciascun milite, poi si procede facendo "muovere" all'Amiga. Sullo schermo vediamo quindi immediatamente i risultati delle mosse, in termini di spostamento dei pezzi avversari o di caduti sottò il fuoco nemico. Alla fine di ogni missione si ottiene una completa analisi statistica del nostro comportamento (che può fruttare riconoscimenti) e di quello dei partecipanti (che possono ricevere delle promozioni).

La tecnica

Tutto è visto dall'alto, con i personaggi individuabili come omini in risolvenza minima.

La grafica di fondo è anche meglio, così come gli effetti sonori, in parte digitalizzati ottimamente. Bisogna comunque tenere conto che non si tratta di un Arcade puro, ma di un gioco strategico "da tavolo", quindi il livello di intelligenza e strategia prevista dai programmatori è certo più importante della grafica.

Il voto

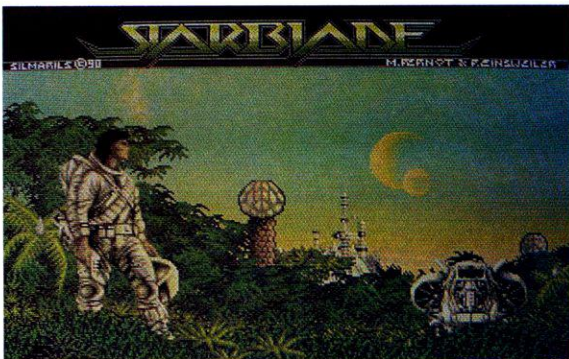
Abbastanza ben fatto, ma la grafica ha qualche neo perchè poteva certamente essere sviluppata con qualche attenzione in più. 8+.

L'atmosfera dei film di fantascienza degli anni '50 è rievocata nelle videate del gioco

Computer: Amiga inespanso
Gestione: Joystick/tastiera
Tipo: Arcade multifase
Softhouse: Palace

STARBLADE

Anno 3001, incipiente guerra tra mondi in un universo dove le distanze sono incredibilmente brevi e le forme di vita note altrettanto strane.

Il gioco

Si rivestono i panni del mercenario, pilota della aereonave **Starblade**, dove è possibile accedere ai magazzini, alla sala motori ed ai computer di bordo. Dal pannello di controllo si ha il completo controllo di tutte le parti dell'astronave ed è quindi possibile stabilire la rotta, ricevere messaggi, variare i modi di viaggio (iperspaziale o convenzionale) e tutto quanto un buon ammiraglio di aereonave sa. Dopo avere esplorato la Starblade, si può iniziare a visitare pianeti.

Prima di scendere sulle loro superifici è però necessario scegliere un'opportuna attrezzatura difensiva, dato che spesso ci si ritrova faccia a faccia con insettoni e alieni vari assortiti, talvolta assai poco amichevoli. Taluni pianeti possono essere esplorati solo se si possiede una sufficiente riserva di ossigeno, particolare che deve essere desunto da un'analisi attenta delle info di ogni pianeta fornite dal computer di bordo.

Ovviamente, il fascino del gioco è proprio nella varietà di pianeti e nella loro disuniformità: dal deserto alla città industrializzata sono tutte scene fantasiose e suggestive.

La tecnica

La presentazione di Starblade è di grande effetto: la nave e gli sfondi sono eccellenti.

L'atmosfera, chiaramente ispirata ai film di fantascienza di serie B degli anni '50, è resa molto bene.

Gli alieni sono un po' troppo poco nitidi, mentre lo scrolling dello schermo è sfarfallante, particolare che ricorda comicamente proprio i film degli anni 50 prima citati!

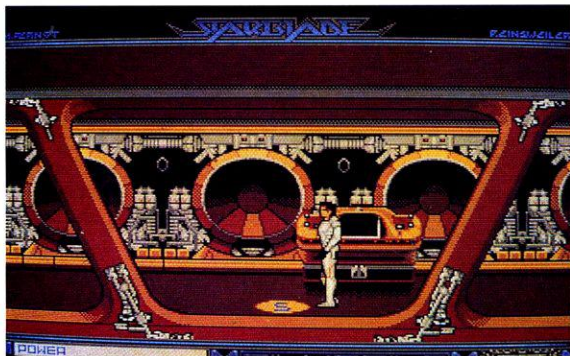
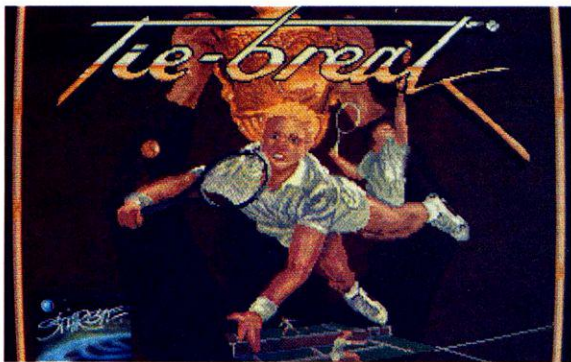
Gli effetti sonori sono limitati a qualche crepitio di laser e poco altro.

Il voto

Grande varietà di scenari, grafica decente, discreti incentivi. 6 1/2.

TIE BREAK

Una simulazione del gioco del tennis davvero realistica ed avvincente

Computer: Amiga inespanso
Gestione: Joystick
Tipo: Simulatore Tennis
Softhouse: Ocean

L'ennesima simulazione del gioco del tennis, con la variazione della prospettiva rigidamente perpendicolare dall'alto ed un buon numero di opzioni iniziali.

Il gioco

Tie Break consente di provare i brividi del racchettare in tutte le situazioni possibili per i grandi giocatori: dal muoversi sulla insidiosa erbetta verde dell'Open di Wimbledon, al fondale sintetico per corridori della Coppa Davis.

Inizialmente si può scegliere se giocare da soli o come parte di un gruppo di sedici giocatori, dei quali si può affidare il controllo al computer, ovvero si può giocare nei panni di 1, 2 o più giocatori contemporaneamente. Quindi si può scegliere il tipo di torneo ed il tipo di racchetta. Durante la partita il controllo del colpo avviene muovendo il joystick in una posizione obliqua al momento del contatto della racchetta con la palla: maggiore è la durata del contatto della palla con la racchetta insieme alla posizione obliqua della levetta del joystick, maggiore è la forza del colpo, mentre lasciando rientrare al centro la levetta del joystick, opportunamente si dosa la direzione di invio della palla. Il tasto di fuoco serve a colpire con maggiore violenza la sfera, nel caso si vogliano eseguire delle schiacciate.

La tecnica

Una delle maggiori forze del programma è l'atmosfera realistica che crea. Le animazioni sono molto graziose, così come le figure dei tennisti.

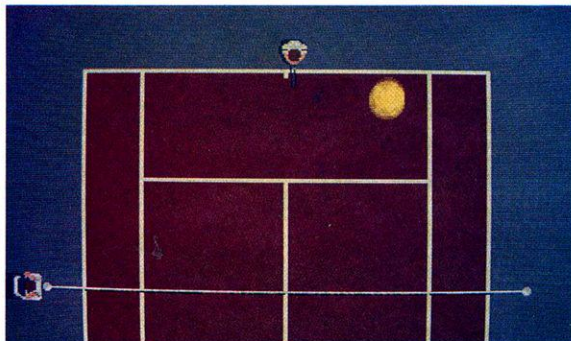
Gli effetti sonori, per quanto limitati dal tipo di gioco, sono estremamente realistici, in particolare i contatti delle racchette con la pallina e le ovazioni del pubblico. La prospettiva dall'alto conferisce una nota di interesse ad un programma altrimenti destinato ad essere troppo simile a molti altri.

Il voto

Buona atmosfera, buona tecnica, molte variazioni ed incentivi. 7 1/2

NEUROMANCER

Avete perso una specie di Bancomat; con conseguenze più che drammatiche

Computer: amiga inespanso
Gestione: Mouse
Tipo: Avventura animata
Softhouse: Electronic Arts

Un Role Playing Game molto tetro, ma variegato; ispirato dal libro di **William Gibson**.

Il gioco

Anno 2058: la tecnologia è progredita al punto che tutte le aziende e le associazioni sono connesse tra loro tramite un enorme network computerizzato. Si gioca nel ruolo di una sorta di cowboy che ha perso l'accesso al ciberspazio, il proprio denaro, il proprio lavoro e, insomma, è decisamente jellato. Si girovaga inizialmente per **Chiba City**, un'area della Tokyo del futuro che è considerata la capitale tecnologica del mondo. Via mouse si effettuano gli spostamenti, si afferrano ed esaminano gli oggetti, che compaiono nelle schermate grafiche (alte metà schermo) presentate costantemente sullo schermo. L'interazione è semplificata, ma è necessario conoscere l'inglese per decifrare il testo che accompagna le scene e le azioni da noi intraprese, nonchè i **fumetti** con i quali dialogano gli altri personaggi.

La Tecnica

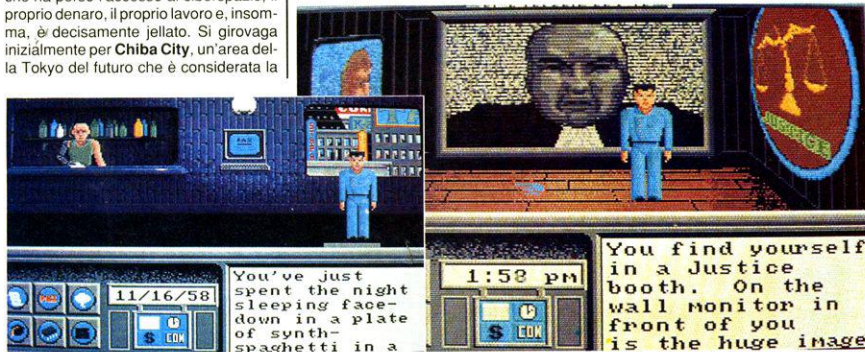
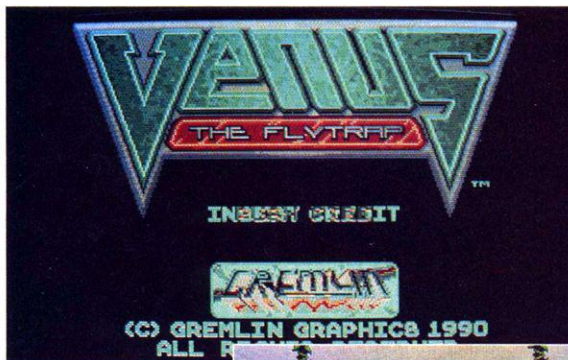
L'atmosfera pesante e dimessa del libro di Gibson viene, nel gioco, resa perfettamente. Ovviamente, dato il basso numero di scene animate, la rievocazione del libro non può essere troppo fedele per chi lo ha letto, causa la mancanza dei colpi di scena e delle scene movimentate, nonchè di alcuni particolari architettonici citati nel volume, come i vecchi edifici camuffati con nuove facciate.

In ogni caso, il numero di rompicapo inserito è elevato e l'interazione via mouse con il giocatore molto ben riuscita. Gli effetti sonori sono invece carenti.

Il voto

Buona grafica, ottimi incentivi, trama appassionante. 8+.

VENUS

Quando la natura viene distrutta, il rimedio escogitato può essere peggiore del male

Computer: Amiga inespanso
Gestione: Joystick
Tipo: Scrolling zapper
Softhouse: Gremlin Graphics

L'uso indiscriminato dei pesticidi ha causato uno squilibrio totale nell'ecosistema, sterminando tutti gli insetti.

Per ripristinare la situazione, la **Gremlin Graphics** ha pensato bene di produrre insetti cibernetici, che però danno filo da torcere...

Il gioco

Venus "the flytrap" è l'equivalente, per gli insetti nocivi, di Robocop per gli umani. Lungo le dieci aree di gioco (dalla Foresta Proibita al Mondo Arboreo) il compito è di muovere Venus lungo lo scenario "left-scrolled" tra insetti, cannoncini laser ed architetture straordinariamente complicate, per raccogliere punti di bonus ed uccidere gli insetti sfuggiti al controllo.

Non si tratta, comunque, di uno scrolling zapper puro, alla Defender per intenderci.

Talvolta, infatti, ci si trova a dovere affrontare sia delle situazioni puramente "tuttogrilletto" (dei sottogiochi ad ogni fase), sia a risolvere dei piccoli rompicapo per passare indenni alcune situazioni.

La tecnica

Un gioco molto ben presentato nell'introduzione, che si svolge in un programma dalla grafica estremamente colorata, tramite una tecnica di gestione del Copper certamente più evoluta della media dei programmi.

Lo scrolling è fluidissimo, così come le animazioni. Gli sprites sono ricchi di dettagli.

La musica è forse stucchevole alla lunga, ma comunque ben realizzata.

Il voto

Non originalissimo, ma realizzato molto bene tecnicamente. 8.

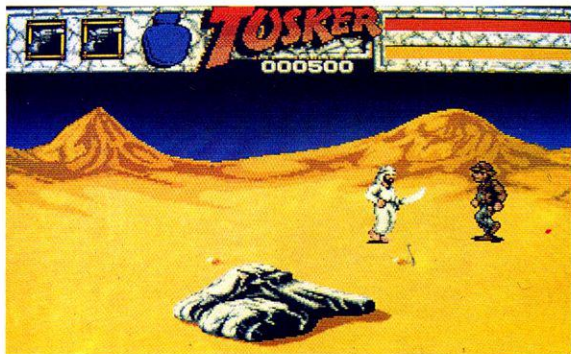
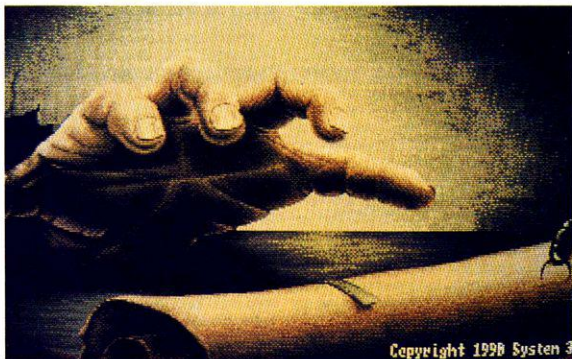
Il cimitero degli elefanrti, e le ricchezze che rappresenta, vi aspettano in Africa

Computer: Amiga inespanso
Gestione: Joystick
Tipo: Arcade multifase
Softhouse: Activision

TUSKER

Secondo la leggenda, esiste un cimitero degli elefanti da qualche parte dell'Africa. Andiamo alla ricerca del prezioso avorio?

Il gioco

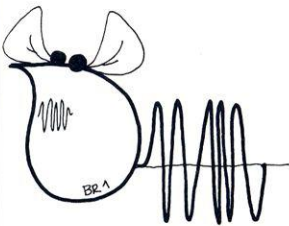
Tusker è ambientato nel deserto; il nostro eroe è quindi in perfetta tenuta "coloniale" moderna con pistola, un po' alla Indiana Jones. Lo scopo è di ricercare, con alcune procedure da adventure (come procurarsi le torce per visitare le caverne ed i proiettili per abbattere gli avversari, che però ci affronano stile arcade!), il cimitero degli elefanti, che il nostro babbo ha cercato per tanto tempo e che ora, avendo tirato le cuoia, tocca a noi ricercare. I nemici sono arabi dotati di scimitarre assetate di sangue, serpenti a sonagli e altre stranezze che vi lasciamo la sorpresa di scoprire da voi.

La tecnica

Scrolling rigidamente orizzontale per questo gioco, dotato di sprites piuttosto nitidi e realistici, ben animati, eccettuate le situazioni nelle quali il nostro eroe deve fare qualcos'altro che camminare e sparare (ad esempio, circumnavigare un ostacolo), dove si evidenziano alcune pecche. La grafica di sfondo è molto curata, stile cartoons, sebbene supercompressa e generata algoritmicamente, come prevedibile data l'ampia scena di gioco.

Lo scrolling, purtroppo, non è continuo, perciò il nostro eroe non può vedere che cosa lo aspetta, potendo a volte cercare riparo dai nemici nello "schermo precedente" con un piccolo colpo di joystick.

Gli effetti sonori sono piuttosto scarsi.

Il voto

Buona ambientazione, buoni sprites, scarsi effetti sonori, buoni incentivi. 6+.

di Luigi Callegari

HARD DISK PER AMIGA

Vi sarete accorti, usando l'Amiga con i soli drives per floppy disk, che la velocità di trasferimento dei dati non è proprio esaltante. Con un hard disk, invece...

Chi acquista un'Amiga, lo fa generalmente attratto dalle sue spettacolari capacità grafiche e sonore, unite ad un prezzo d'acquisto certamente molto invitante. In genere, la gran parte degli acquirenti di Amiga opta per l'A-500, che viene spesso relegato alla funzione di sofisticata macchina per videogiochi di lusso.

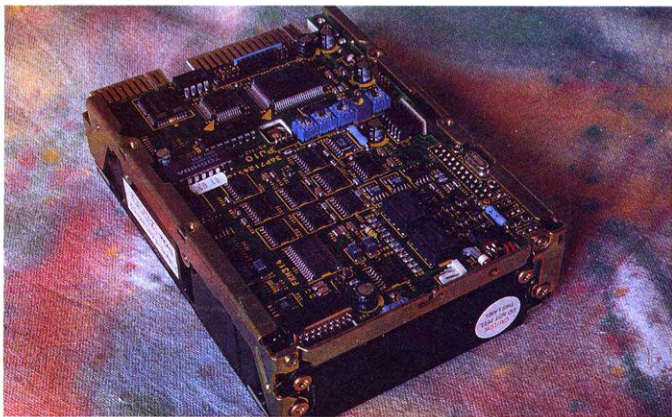
Presto o tardi, comunque, molti vengono spinti dal desiderio di incominciare a fare qualcosa di più con Amiga, magari dopo la laurea o finite le scuole superiori, ed iniziano a provare il dilemma di come investire del denaro in espansioni ed accessori. In effetti, un A-500 o A-2000 con il solo drive interno (e solo 512 / 1024 Kilobytes di memoria) non consentono di svolgere agevolmente alcun lavoro serioso: programmazione con linguaggi evoluti od archiviazione di dati, ad esempio. O meglio, lo si può anche fare (sino a poco tempo fa c'erano anche persone che provavano persino a fare contabilità con un PET od un C/64, macchine certamente molto meno potenti di Amiga!), ma al costo di enormi perdite di tempo e sopportando notevoli disagi, quando effettivamente possibile.

Fortunatamente, oggigiorno in Italia si possono trovare molte periferiche ed espansioni per Amiga. Dopo un secondo drive esterno, ed una espansione ad un megabyte se si possiede un A-500, l'acquisto che certamente può consentire di modificare radicalmente il proprio modo di lavorare con Amiga, aprendo orizzonti di applicazione prima sconosciuti, è il disco rigido.

Hard Disk?

La parola **Hard** Disk nasce contrapposta a **Floppy** Disk, che letteralmente significa "disco molle", e che tutti noi conosciamo nei formati 5,25" (quello del C/64 e di molti MS- DOS) e 3,5" (quello di Amiga, con il guscio di plastica dura, comune ora anche agli MS-DOS).

Il nome sta ad indicare non già una unità di memoria basata su di un disco "duro" (come sta scritto erroneamente sulla confezione degli A-590...), ma un disco "rigido", cioè non disinseribile dalla confezione entro cui viene sigillato dalla fabbrica produttrice.

Nell'Hard Disk una testina magnetica corre, perfettamente allineata una volta per tutte in fabbrica, lungo l'asse longitudinale del disco, mossa da precisissimi motorini controllati dal sistema operativo,

per leggere o scrivere dati. I dischi rigidi, esattamente come i floppy, sono infatti in grado di conservare i dati, ma anche di cancellarli e riscriverli.

La superiore affidabilità degli Hard Disk e l'alta capacità di memorizzazione (si va dai 20 ai 600 e più megabytes), dipendono dalle tecnologie di costruzione: in fabbrica vengono usati supporti e materiali migliori di quelli dei floppy, particolari tecnologie di assemblaggio e verifica che implicano l'uso di personale specializzato e sofisticati macchinari, dai quali derivano ovviamente un certo costo finale della periferica.

Oltretutto, gli Hard disk consentono una velocità di gestione dei dati assai superiore a quanto possibile con i comuni Floppy Disk, come si può leggere dalla tabella riportata, in cui sono riportati i risultati dei **Benchmarks** eseguiti con un apposito programma di pubblico dominio (**DiskSpeed**).

Il motivo principale della maggior velocità degli HD risiede nel fatto che i dischi, in questi, ruotano più rapidamente che nei FD, sia perchè le testine di lettura / scrittura non toccano la superficie del disco, sia perchè, nei floppy disk, i maggiori margini di tolleranza oseervati in fase di fabbricazione (delle testine e dei dischetti stessi) impedirebbero alte velocità, pur se realizzabili.

Amiga e HD

Quando si formatta un disco, sia esso floppy o rigido, tramite la testina del drive si incidono le informazioni necessarie al sistema operativo del computer per utilizzare il disco stesso. Con il comando **Format** impartito con appositi parametri (da **Shell**) o **Initialize** (da Workbench), Amiga provvede a scrivere sul disco i **cilindri** ed i **settori**. I cilindri sono così chiamati perchè rappresentano delle "marcature" (ovviamente a livello magnetico), paragonabili vagamente a quelle incise sui dischi microsolco, con la differenza che queste ultime sono un'unica spirale, mentre quelle dei dischetti sono tante circonferenze concentriche.

I "cilindri" sono fisicamente separati, in quanto sulla superficie del disco non si incrociano mai, mentre i **settori**, o **blocchi**, sono suddivisioni logiche, in quanto sono materialmente realizzati dal sistema operativo scrivendo, nei cilindri, particolari informazioni che li delimitano all'interno di suddivisioni fisiche continue.

I normali dischetti di Amiga prevedono, per ciascuna facciata, 80 cilindri, ciascuno suddiviso in 11 blocchi contigui di 512 bytes ciascuno.

La capacità totale di un floppy risulta quindi di...

512 x 11 x 80 x 2 = 901.120 bytes

...che, in effetti, risultano "solo" 880K circa perchè il sistema ne usa una parte per gestire il disco, appunto per le suddivisioni logiche prima dette.

Per formattare un disco rigido è normalmente necessario conoscere i suoi parametri interni, che possono variare da tipo a tipo. In genere, occorre sapere almeno il numero di **testine**, il numero di settori e di blocchi, inserendoli o in un punto del file **devs:mountlist** sul dischetto del Workbench, oppure direttamente in una posizione protetta dell'Hard Disk, tramite programmi di installazione appositi forniti con tutti gli apparecchi di regolare distribuzione in Italia.

La formattazione può anche dividere in varie **partizioni** il disco rigido. Una partizione è una suddivisione logica, non fisica, dell'hard disk. In pratica, si può dare al sistema operativo l'illusione di avere collegati più hard disk (di nome **DH0**, **DH1**, **DH2**...) mentre, in effetti, la meccanica è una sola. Ovviamente, se si possiede un HD da **40 megabyte**, ad esempio, si potrebbero avere due partizioni per raggiungere complessivamente questo valore in qualsivoglia suddivisione: 10+30, oppure 10+10+10+10, e ancora 20+20, eccetera.

Partizionare un HD può consentire, tra l'altro, di suddividere e gestire con maggiore efficienza lo spazio disponibile, in particolare quando è molto ampio. Inoltre, garantisce una maggiore sicurezza nel caso di operazioni accidentali o danneggiamenti dell'HD, in quanto i dati delle diverse partizioni sono più "divisi" e quindi gestibili e recuperabili (o danneggiabili!) separatamente.

I **controller** moderni, ovvero i circuiti elettronici che fanno da interfaccia tra la meccanica dell'Hard Disk ed il computer (come quelli montati all'interno dell'Hard Disk siglato **A-590** e dell'Impact GVP 500 per Amiga 500 e dell'A-2091 per A-2000) eseguono il cosiddetto **automounting** della periferica. Ciò significa che, all'accensione, Amiga riconosce subito la presenza di un HD, senza bisogno di inserire appositi comandi nella startup-sequence del dischetto in DF0, ed eventualmente esegue la procedura di installazione usan-

do direttamente i files sull'HD. Tale caratteristica viene in gergo chiamata **Autoboot** ed era assente nei primi Hard Disk per Amiga 1000 e Amiga 2000.

Il controller **A-2090A** per A-2000, invece, era in grado di eseguire l'autoboot, ma usava una procedura particolare in quanto non poteva eseguirlo da una partizione formattata col nuovo **FFS** (**F**ast **F**ile **S**ystem), previsto a partire dalla versione 1.3 del sistema operativo. Dal momento che questo sistema di filing consente una maggiore efficienza nella gestione dell'HD, viene usato correntemente, ma il controller Commodore non era in grado di eseguire un autoboot da una partizione formattata in questo modo, quindi doveva eseguire il boot da una minipartizione formattata in modo normale.

Attualmente, tutti i controller moderni, sia di marca Commodore che di altre case, possono eseguire l'autoboot da partizioni FFS direttamente. Si rammenti, però, che l'autoavviamento è possibile solo se si possiede il Kickstart V1.3 (o superiori) nel proprio computer.

Device: **DF0:**
Comment: **Floppy Disk** Commodore standard, con meccanica Chinon.
0 Files/s Create ; Creazione files
1 Files/s Open/Close ; Apertura e chiusura
3 Files/s Scan ; Scansione
1 Files/s Delete ; Cancellazione
2 Seek/Read ; Posizionamento e Lettura

Buffer Size	**512**	**4096**	**32768**	**262144**
Bytes/s Create	4722	5008	4970	4738
Bytes/s Write	3077	3085	3175	3044
Bytes/s Read	11548	6242	12725	11773

Device : **DH0:**
Comment: Hard Disk GVP "**Impact 500**" per Amiga 500
7 Files/s Create
15 Files/s Open/Close
56 Files/s Scan
17 Files/s Delete
196 Seek/Read

Buffer Size	512	4096	32768	262144
Bytes/s Create	27860	89181	91522	90086
Bytes/s Write	29199	126352	152411	158415
Bytes/s Read	29213	127745	155428	159925

Device: **DH0:**
Comment Hard Disk Commodore "**A590**" per Amiga 500
5 Files/s Create
9 Files/s Open/Close
42 Files/s Scan
4 Files/s Delete
187 Seek/Read

Buffer Size	**512**	**4096**	**32768**	**262144**
Bytes/s Create	19715	69888	95418	134706
Bytes/s Write	21375	96890	172184	182842
Bytes/s Read	21388	97045	172562	183159

Benchmark
Risultati ottenuti con il programma **DiskSpeed** *sul floppy disk standard di Amiga e su due Hard Disk per A-500. I primi cinque parametri indicano quante delle operazioni indicate al secondo vengono svolte dalla periferica operando sui files. Il quinto valore quante volte al secondo, al massimo, viene spostata la testina per una nuova lettura. Le tabelle indicano quanti bytes al secondo vengono gestiti dalla periferica durante la creazione, la scrittura e la lettura dal disco*

Pregi

Per l'Amiga 500 esistono diversi modelli di HD, i quali consentono, normalmente, di espandere anche la memoria di Amiga. Bisogna infatti precisare che un HD usa una certa quantità di **Ram** del computer, che diventa ragguardevole se il proprio sistema è dotato di un solo megabyte. Dopo l'acquisto dell'HD, in genere, ci si preoccupa di espandere anche la **Fast Ram**, in modo da usare con la massima efficienza il proprio Amiga.

Oltretutto, i prezzi delle Ram sono talmente ribassati che espandere anche di un paio di megabyte il proprio Amiga, avendo gli HD che lo consentono tramite appositi spazi interni, costa meno dell'acquisto di un floppy disk drive esterno.

Ad esempio, l'A-590 Commodore consente di inserire, al suo interno, vari chip di memoria per aggiungere 512 K, 1 oppure 2 megabytes di memoria (utilizzando chips da 256K x 4 CMOS), mentre l'Impact GVP A-500 prima serie consente di aggiungere 2 oppure 4 megabytes (tramite chips di tipo SIMM) e, la nuova serie appena uscita negli USA, addirittura raggiungere il limite di 8 megabytes di espansione Ram.

Ovviamente, la memoria è di tipo Fast Ram, ed è "vista" automaticamente dall'A-500 al momento dell'accensione, ovvero ne viene eseguito il cosiddetto (impropriamente) "automounting", se si tratta di memoria "autoconfigurante", come dicono i tecnici. Vecchie espansioni di memoria per Amiga 1000 e Amiga 2000, invece, imponevano la presenza di istruzioni nella **startup-sequence** per rendere utilizzabili al sistema le espansioni di memoria. In quest'area **Exec** pone i propri dati (le "variabili di sistema", per usare

un termine del mondo ad otto bit), invece che nella Chip Ram come al solito, migliorando le prestazioni effettive del computer, in quanto l'accesso ai dati in Chip Ram da parte del microprocessore deve avvenire più lentamente, a causa dell'accesso contemporaneo dei chip custom presenti in essa.

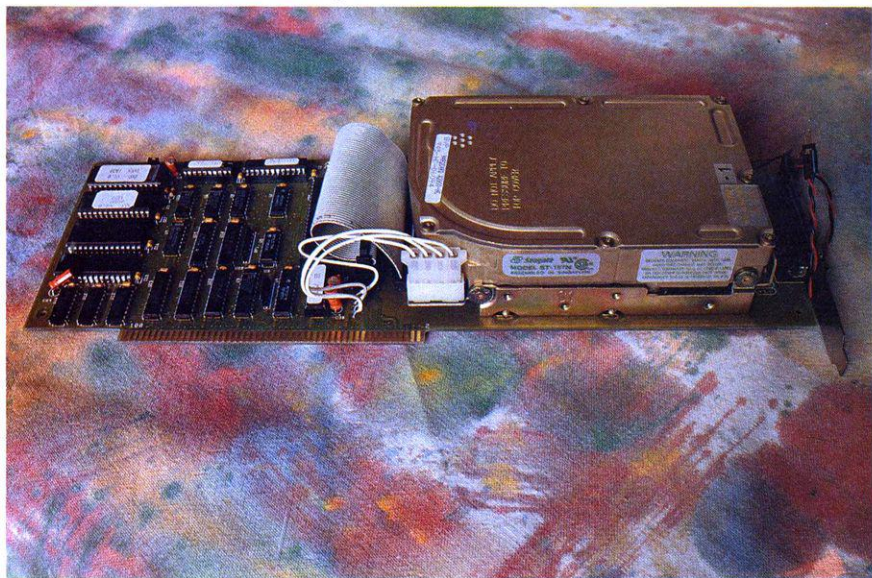
Per quanto riguarda l'Amiga 2000, le schede HD si sprecano. Esiste la nuova **A-2091** Commodore, che ha sostituito la "vecchia" e gloriosa A-2090A, poi una marea di altre schede, a volte prodotte da "cantinari", di qualità variabile. Alcune schede, come le **GVP** e le **Supra**, consentono addirittura di montarvi l'HD, oltre all'espansione di RAM, rendendo inutile l'acquisto di una A-2058 per espandere la memoria dell'A-2000.

Bisogna comunque informarsi bene prima dell'acquisto, soprattutto per le schede poco note, perchè a fronte di piccoli risparmi si potrebbero avere problemi di incompatibilità con altri accessori e periferiche aggiunte all'A-2000 successivamente: schede velocizzatrici, espansioni di Ram, digitalizzatori eccetera.

Lavorare con l'HD

Come dicevamo, avere un Hard Disk collegato all'Amiga ne aumenta incredibilmente le potenzialità. Non è più necessario avere una marea di dischetti sul tavolo per caricare ora un editor, ora una utility, ora un file di archivio. Tutto ciò che serve comunemente è subito accessibile, rapidamente, senza lavori da Disk Jockey.

In particolare, chi usa linguaggi di programmazione, ma anche chiunque usi estesamente programmi di grafica e, insomma, sfrutti ad un certo livello il proprio Amiga, avrà un beneficio incredibile. Ad esempio, gestire database cospicui, scrivere un libro con molti capitoli, manipolare immagini digitalizzate, sono operazioni (forse) eseguibili anche usando un sistema dotato di soli floppy, ma diventano estremamente più pratiche se si possiede un HD. Si può stare certi che, una volta iniziato a lavorare con un HD, ritornare a lavorare con i dischetti è un supplizio inenarrabile. Alcuni programmi, durante la fase di debug, mandano spesso Amiga in crash e costringono l'utente ad un reset. Con un HD, in casi come questo, occorrono una dozzina di secondi soltanto per ricaricare il sistema operativo, contro la marea di tempo richiesta per un riavviamento da floppy.

Inoltre, con la memoria espansa, si possono rendere residenti numerosi programmi contemporaneamente (editor, files di libreria, programmi di ritocco grafici), aumentando enormemente la velocità di lavoro e riducendo i tempi di attesa. Questi, a causa della velocità di accesso dei floppy di Amiga e della lunghezza dei files, spesso fa ricordare i tempi in cui si caricava da cassetta con il C/64!

di Alessandro de Simone

UN MODEM CHE PARLA ITALIANO

Chi desidera utilizzare apparecchi e schede per PC si imbatte, troppo spesso, in una manualistica scritta in inglese. Il prodotto di queste pagine offre libretti di istruzione completamente tradotti in italiano

Esterno o interno? Questo è l'amletico dubbio di chi si accinge ad acquistare un modem per lanciarsi a capofitto nell'affascinante mondo delle telecomunicazioni via computer. Di (quasi) ogni modello, infatti, vengono offerte due versioni che presentano vantaggi e svantaggi. Se si hanno due computers, ad esempio un **PC compatibile** ed un **Amiga**, è certamente consigliabile la versione esterna che, grazie ad un cavo **Rs-232**, consente di applicare alternativamente il modem ad uno dei due elaboratori (stessa cosa se, nella casa in montagna o al mare, si ha un computer "di fortuna").

Allo stesso modo, chi vuole vedere lo stato di funzionamento del modem attraverso l'accensione dei led, deve rinunciare alla versione interna; chi dispone di poco spazio sulla scrivania, però, si vede costretto a rinunciare al modello esterno, e così via.

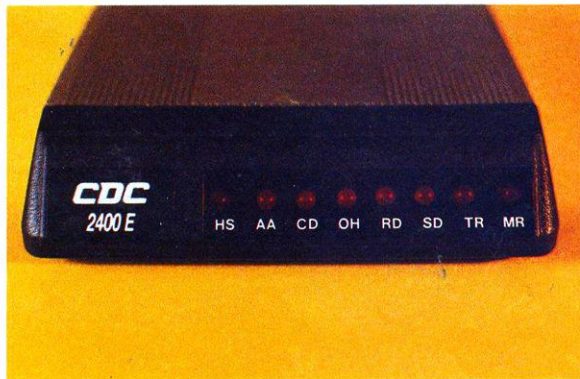
L'importante, comunque, è che il prodotto sia valido ed affidabile, soprattutto perchè dovremo operare con le linee telefoniche italiane della SIP (e chi ha orecchie per intendere...).

Scherzi a parte, il notevole progresso tecnologico consente, all'utente odierno, di disporre di apparecchiature sofisticate in grado di reggere ritmi di lavoro non indifferenti.

Non tutti sanno, infatti, che la miniaturizzazione dei componenti elettronici non è dovuta al desiderio di offrire apparecchi sempre più piccoli e meno ingombranti.

Il motivo principale, invece, risiede nel fatto che un circuito miniaturizzato offre una velocità di elaborazione maggiore, richiede un'energia elettrica di minore entità che, a sua volta, genera un quantità di calore più bassa ed evita il pericolo di surriscaldamento dei circuiti, le cui conseguenze sono facili da immaginare.

Oggi bastano pochi (e piccoli) circuiti integrati per svolgere funzioni che, qualche anno fa, richiedevano un maggior numero di chips, una maggiore potenza elettrica ed una selezione dei componenti rigorosa.

Per fortuna, oggi, è possibile inserire su una scheda elettronica di ridotte dimensioni non solo il modem vero e proprio, ma anche quella circuiteria di controllo e di verifica in tempo reale che rendono, di fatto, un modem moderno

molto più sicuro di quelli offerti tempo addietro.

Gli standard

La caratteristica principale di un modem, che quasi sempre condiziona la scelta dell'utente, è rappresentata dalla velocità di rice / trasmissione dei dati.

Non ci stancheremo mai di ripetere, infatti, che non ha senso risparmiare qualche biglietto da 10.000 lire per procurarsi un modem di minor potenza; e questo sia perchè le spese principali di un modem sono rappresentate dagli incrementi delle bollette telefoniche (maggior velocità = minor durata del tempo di collegamento = minor numero di "scatti" SIP), sia perchè i modem più potenti dispongono di una circuiteria più sofisticata, che consente di effettuare controlli non disponibili su schede di minore contenuto tecnologico.

Gli standard di comunicazione universalmente accettati sono numerosi ma, di certo, i più noti sono quelli cosiddetti **Hayes Compatibili**.

Il modem **CDC 2400**, fornito nelle versioni interna (Ms-Dos) ed esterna (universale) con manuale interamente tradotto in italiano (e, per di più, tradotto abbastanza bene; caratteristica purtroppo rara da riscontrare in prodotti di basso costo...) offre caratteristiche tecniche adeguate ai compiti che deve svolgere:

Modi operativi: asincrono, sincrono (ma solo per modelli E ed ECE); full / half duplex; chiamata in originate automatica o manuale; ripsosta in answer automatica o manuale; protocolllo MNP 5 (modelli ECE, ICE, CEPS, CELT).

Compatibilità: CCCITT **V. 22 bis**; CCITT **V.22**; CCCITT **V.21**; Bell 212 A; Bell 103. Velocità dati sincroni (solo per modelli E ed ECE): 2400 bps (bit per secondo) +/- 0.01 % (V.22 bis). 1200 bps +/- 0.01 % (V.22).

Velocità dati asincroni : **2400**, **1200**, **0-300 bps** +/- 1.25%

Formato caratteri (in modo asincrono): massimo 10 bit, compresi Start, Stop, Parità.

Sorgente del clock: interna (generata dal modem); esterna (generata dal DTE). Altre caratteristiche, riportate sul manuale, rendono appetibile la periferica e sono state sperimentate nella realtà.

L'esemplare consegnatoci in visione, infatti, è stato applicato alla nostra banca dati per cinque giorni di seguito (24 ore su 24) e non ha mai presentato problemi di sorta.

Il software

Non tutti sanno che un modem è possibile programmarlo (perfino in Basic!) con la massima semplicità, dal momento che, al suo interno, è presente un circuito in grado di decodificare normalissime stringhe Ascii che rivestono particolare significato.

Ad esempio, inviando al modem la stringa **ATM0** si impedisce al monitor audio di essere attivo (con **ATM2**, invece, lo si rende sempre attivo).

Con **ATB0** si rende operativo lo standard CCCITT V.22; con **ATB1**, invece, lo standard Bell 212A; e così via.

Tutti i comandi accettabili dal modem, ripetiamo, sono indicati sul manuale. Chi lo desiderasse, pertanto, può addirittura scrivere, da solo, un completo programma di comunicazione.

Modem	Interno	Esterno
Prezzo	Minore	Maggiore
Ingombro	Nullo	Evidente
Controllo Led	Impossibile	Verificabile
Accensione	Automatica	Manuale
Interfaccia	Interna (su Bus)	Cavo esterno Rs-232
Alimentazione	Interna (su Bus)	Esterna
Uso con più computers	Impossibile	Possibile

Principali differenze tra modelli interni ed esterni

Chi, invece, di programmazione non sa un tubo (o, più eufemisticamente, non ha il tempo di occuparsene...) e desidera solo collegare il modem e scambiar dati via telefono, non deve fare altro che utilizzare il dischetto, incluso nella confezione, che gestisce in modo totale il CDC 2400.

Si tratta del diffuso programma **BitCom** che, operando nello standard Ms-Dos, è in grado di sfruttare a fondo tutte le caratteristiche del modem di cui ci stiamo occupando.

Naturalmente, dal momento che non si tratta di un programma specifico per il modem in questione, è possibile utilizzarlo anche con altri apparecchi, eventualmente in possesso dell'utente, purchè di standard Ms-Dos.

BitCom è il programma originale, in inglese. Il manuale, invece, è totalmente in italiano; nelle oltre **100 pagine** di cui è composto, passa in esame, una per una, tutte le videate che compaiono nelle varie fasi di utilizzo.

Naturalmente non si limita a dire come settare i vari parametri, ma anche il perchè. Ciò farà piacere a quella categoria di utenti che non si accontentano di operare passivamente, ma desiderano approfondire determinati argomenti.

Il prezzo

Al momento di andare in stampa è giunta, in Redazione, una telefonata che ci invitava ad omettere il prezzo indicatoci al momento della consegna dell'esemplare provato.

E' molto probabile, infatti, che quando leggerete queste righe sarà stato perfezionato un particolare accordo commerciale che favorirà un notevole abbassamento dei prezzi di listino al pubblico. Non ci rimane che concludere ricordando che la **Computer Discount** ha sedi a Bologna, Firenze, Genova, Milano, Modena, Palermo e Pisa.

Ringraziamo il sig. Pierangeli della sede di **Milano** (Via Cenisio 12, tel. **02 / 33.10.02.04**) che ha consentito la prova del modem e che ci ha pregato di segnalare, ai nostri lettori, la possibilità di procurarsi il modem anche per corrispondenza.

di Domenico Pavone

METTI UN PC NEL TUO A- 500

Chi ha voglia di entrare nel mondo Ms - Dos spendendo poco denaro, da oggi può prendere in seria considerazione l'acquisto di una scheda piccola piccola, che rappresenta, però, un vero e proprio computer PC compatibile

Solo due mesi fa, nelle pagine di questa stessa rubrica, si accennava in toni quasi entusiastici alla continua riduzione del gap che divide due "macchine" così vicine eppure così diverse, non fosse altro che per il differente impatto (sempre traumatico) sulle tasche dell'utente finale: l'Amiga 500 e la "cugina" Amiga 2000.

In quell'occasione si dava il benvenuto al proliferare di nuove espansioni di memoria al di sopra dei soliti 512 KRam. Stavolta un altro baluardo viene a cadere, finorá esclusivo dominio dei modelli superiori: l'apertura verso il mondo **Ms-Dos**.

E non si sta parlando di giocattoloidi come **emulatori software** utili, tutt'al più, per dare un'occhiata al modo di operare dei Personal, ma della possibilità di trasformare Amiga500 in un VERO Personal XT via hardware, senza rinunciare a quelle caratteristiche che ci hanno fatto scegliere un computer come Amiga.

Anzi, come vedremo meglio tra breve, potenziandone addirittura il corredo di Ram. E, cosa di non secondaria importanza, con una facilità disarmante, **senza alcun intervento** sulla componentistica di Amiga.

Non sembra vero di poter rispondere... per le rime ai più fortunati cugini-amighi che potevano vantare l'uso di schede come le **Janus**, strettamente riservate ai 2000!

Ma bando alle ciance, è venuto il momento di fornire nome e cognome (forse paternità sarebbe più adatto) al nuovo.. oggetto di desiderio: **Power Pc Board**, una scheda hardware prodotta dalla olandese KCS.

Senza mezzi termini, si tratta di un vero secondo computer, fornito di un proprio microprocessore (**Nec V30** notoriamente migliore dell'**8088** e, con questo, totalmente compatibile), di una propria **memoria Ram**, e di tutto il necessario per accedere senza problemi al software dell'ambiente Ms-Dos, per di più gestendo direttamente i(l) drive di Amiga, che diventa(no) in tutto e per tutto **Pc - compatibili**.

Ma procediamo con ordine. Intanto la scheda, dalle dimensioni molto compatte, si inserisce senza alcuna possibilità di errore (è agibile praticamente una sola posizione) nello slot presente all'interno del cassetto riservato alle espansioni di memoria, sulla superficie inferiore di Amiga 500. Naturalmente con i componenti rivolti verso l'interno della macchina e (non ci crederete...) a computer spento.

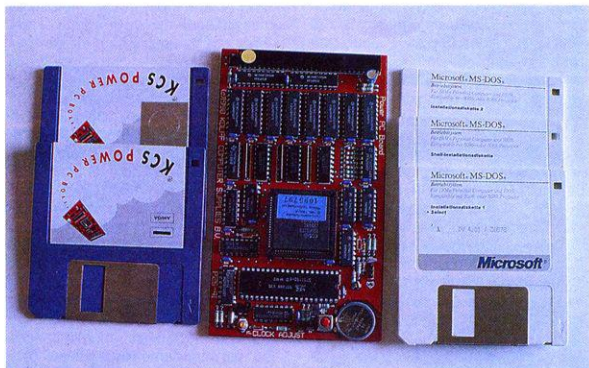
Tutto qua, l'installazione è conclusa. Quella hardware, almeno.

Con un occhio ad Amiga

Se a questo punto si dà vita (diciamo tensione, senza esagerare con l'animismo) ad Amiga, apparentemente non accade nulla di rilevante: senza un disco di boot nel drive, apparirà sul video il ben noto (brutto) disco-con-mano che invita all'inserimento del floppy Workbench. Il solito Amiga, insomma.

Eppure qualcosa, in realtà, è cambiato, indipendentemente dalla agibilità all'Ms-Dos.

Qualcuno forse si sarà chiesto come si possono incrementare i miseri 512KB di memoria base di A500, con la porta di espansione già occupata dalla PC Board.

In vendita presso
FLOPPERIA
Viale Monte Nero,31
20135 MILANO
tel.02-55180484
KCS POWER PC BOARD
(Versione XT oppure AT)

Ebbene, ecco un punto di forza della scheda, tale da giustificare tra l'altro il suo prezzo forse non irrisorio: utilizzando l'Amiga nel suo modo consueto, la **scheda diventa un'espansione di memoria**! Per la precisione, fornisce al nostro beneamato **512 KB di ram** aggiuntiva, ed un **orologio** hardware alimentato da una batteria al litio (non tampone, come si è abituati a sentire) con autonomia di circa due anni. Ad abundantiam, il "clock" è regolabile tramite un trimmerino ben evidenziato sulla scheda, nel caso dovesse manifestare anticipi o ritardi nella scansione del tempo.

Ma non è ancora finita. Sempre in rapporto all'uso di Amiga in modalità tradizionale, non sono solo 512 i KiloByte di memoria in più implementabili.

Oltre a questi, diventa infatti disponibile anche un'altra grossa fetta di ram, di **altri 512 KB**, sfruttabili come Ram Disk resistente al reset. Come a dire, insomma, che il reale guadagno di memoria è di ben **1 Megabyte**, oltre ai 512 Kappa in dotazione alla macchina! Per disporre di questa seconda, comodissima "feature" della Pc Board, è necessaria, però, una installazione via software del nuovo device, che assumerà il nome **RMD:**, oppure **KcsRam**. Nulla di particolarmente difficile, niente paura.

A corredo della scheda, vengono infatti forniti **5 dischetti**, due dei quali servono proprio per avviare Amiga, configurarlo secondo le proprie esigenze, ed eventualmente farlo "switchare" in modo Ms-Dos.

Uno dei due, siglato appunto "Amiga", provvede anche ad organizzare automaticamente un normale disco di boot in modo da poter sfruttare la nuova Ram Disk, nonchè l'orologio hardware.

Tutto ciò che occorre fare è preparare una copia del disco Workbench (i più esperti potranno utilizzare un proprio disco personalizzato) eliminandone (p. es.) il Notepad, in modo da disporre di un po' di spazio.

Quindi, resettato il computer, lanciare il sistema con il disco **KCS Amiga**, mantenedo premuto il pulsante sinistro del mouse.

Così facendo, si accederà ad un menu che prevede, tra le altre cose, l'installazione (ma anche la rimozione via software) della KCSRam e del clock, semplicemente obbedendo alle "ingiunzioni" provenienti dallo schermo.

Ad operazioni concluse, si disporrà di un disco Workbench che, al momento del lancio, provvederà anche ad attivare la nuova Ram Disk a prova di Guru, e a impostare correttamente data ed ora di sistema.

Inutile dire dei vantaggi legati a questo device virtuale, che, non lo si dimentichi, non toglierà nulla al megabyte di memoria disponibile in modo Amiga, e resiste al reset anche con Kikstart versione 1.2!

Da non sottovalutare anche la possibilità, descritta nel manualetto d'uso della Pc Board, di applicare (o far applicare) un interruttore hardware che disabilita completamente la ram aggiuntiva, per quei rari(ssimi) casi in cui possono verificarsi problemi di compatibilità con software... decrepito. Per chi utilizza con disinvoltura il dos di Amiga, va detto che la gestione via software del nuovo device e dell'orologio sono affidati a due nuovi comandi che vengono inseriti nella directory **C:** dal programma di installazione automatica: **PcClock** e **Ramdisk**. Entrambi possono essere adoperati autonomamente da Shell.

Il primo, esattamente come se si trattasse del comando **Setclock**, segue la sua sintassi (e quindi p. es. **PcClock Opt Load** oppure **PcClock Opt Save**), mentre **Ramdisk** va adoperato senza parametri per installare la periferica virtuale; se questa fosse già presente, la cosa verrebbe riconosciuta, senza alcuna altra conseguenza. Come già detto, comunque, simili operazioni diventano necessarie solo se non si ricorre al disco opportunamente preparato tramite la procedura di installazione, che provvede ad automatizzare il tutto nel più semplice dei modi.

E fu subito Personal

Ma, come ovvio, una scheda come la Pc Board non la si acquista solo per avere una (pur comoda) espansione di memoria ed una Ram Disk.

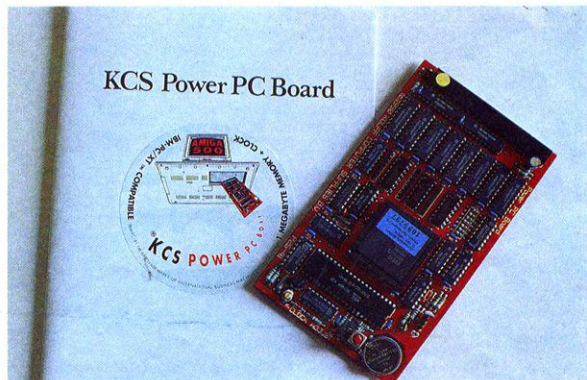
Per vedere il proprio Amiga 500 trasformato in un Pc, basta lanciare lo stesso floppy **KcsPower Amiga**, stavolta senza premere il pulsante del mouse, e dopo un po' apparirà una segnalazione che invita ad inserire il secondo disco, etichettato **Kcs Ms-dos**. A questo punto, se già si possiede un disco di start Ms-dos, lo si può già inserire, e vedere apparire, dopo il boot, il classico prompt di sistema, caratterizzato dal simbolo "**A>**".

La prima volta che si adopera la Pc Board, a meno di non essere già smaliziati con il nuovo ambiente, un tale disco con ogni probabilità non sarà ancora in vostro possesso, ma a corredo della Pc Board è fornita l'ultima versione dell'Ms-Dos, la **4.01**, che consta di 3 dischetti necessari alla sua installazione e relativi **manuali ufficiali della Microsoft**, come a dire la garanzia di poter adoperare da subito l'Amiga "trasformata", godendo di tutte le garanzie fornite da una release ufficiale, ivi compresa la eventuale sosti-

tuzione della manualistica, se non in italiano. Tra l'altro, nel valutare il prezzo totale della Pc Board, non si dimentichi che l'Ms-Dos 4.01 ha normalmente un suo costo: tolto il prezzo di un'espansione per Amiga, e quello del Dos, in fondo non si spende poi troppo... L'installazione dell'Ms-Dos su disco si effettua inserendo il primo dei floppy nel drive interno di Amiga e resettando il computer, ma attenzione: resettandolo come Pc, non più come Amiga, il che equivale alla pressione dei tasti **Ctrl + Alt + Del**. Con questo sistema, il reset riguarderà solo l'ambiente Ms-Dos, mentre il classico Ctrl + Amiga sinistro + Amigadestro farà tornare il sistema al "modo" Amiga. La procedura di installazione, anch'essa completamente automatizzata, va eseguita esattamente come richiesto su qualunque altro computer Pc-compatibile, per cui non ci soffermeremo più di tanto; ma la Pc Board consente anche di configurare il sistema Pc adoperando un programma "intuition-oriented", ovviamente sfruttando Amiga ed il suo comodo interfacciamento grafico. Per accedere a questa utility, è sufficiente operare come prima descritto a proposito dell'instalazione della RamDisk, ovvero adoperare, per il boot, il disco Kcs - Amiga tenendo pressato il pulsante sinistro del mouse, e scegliere l'**opzione 5** (Start Pc Preferences program). Così facendo, è possibile selezionare una serie di scelte, che andranno poi salvate nel disco Ms-Dos di boot.

La maggior parte di queste scelte è di impostazione obbligata, come segnalato nel manualetto specificamente riservato alla scheda, mentre alcune possono essere variate. E' questo, per esempio, il caso del mouse, che può essere inserito facoltativamente in una delle due porte joy. C'è da dire comunque che, com'è prassi in ambiente Pc, il mouse non viene automaticamente implementato, ma necessita di installazione tramite appositi **driver** software. In pratica, un esempio tipico delle differenze esistenti tra i due "mondi", ed alle quali occorre abituarsi approfondendo l'uso dei Pc, se già non lo si conosce.

Il "cuore" Personal del nostro Amiga può naturalmente utilizzare tanto la porta parallela che quella seriale, quest'ultima con un baud rate non superiore ai 4800 bps. La grafica supportata, al momento in cui queste righe vengono scritte, è limitata agli standard **MGA** (schermo bicromatico con risoluzione massima **720 x 348** pixel) e **CGA** (8 colori in **640 x 200** pixel), ma sono già annunciati upgrade per la Pc Board che emulano il **VGA**, standard dotato di maggiore risoluzione e capacità cromatiche. Con gli stessi upgrade dovrebbe anche essere superato un limite della scheda, che al momento non è in grado di gestire gli hard disk, e che dirotta sul drive interno di Amiga tanto la periferica **A:** che la **B:** dell' Ms-Dos. Quando leggerete queste note, considerati gli inevitabili tempi editoriali, gli upgrade dovrebbero comunque essere quasi disponibili. Per il resto, non c'è che da ammirare le prestazioni di Pc Board. Testata severamente, ha "girato" a tutti gli effetti come un XT con frequenza di quasi **8 Mega Hertz**, decisamente buona per una scheda che dovrebbe essere un emulatore. Per superare eventuali problemi di compatibilità software, si può inoltre (con la sola pressione del tasto **F3**) switchare tra una "tipica" disponibilità Ram di 640 KB, ed una più estesa di 704 KB. Per inciso, ai tasti funzione sono collegate anche altre facilitazioni, come il blank di schermo (F4), suono collegato alla pressione dei tasti (F6), colori disponibili (F7), nonchè tutta una serie di regolazioni effettuabili sul momento grazie ad un menu di help richiamabile in qualunque momento con la pressione dei tasti Amiga + Help. Ora e data di sistema, e per sistema si intende stavolta il Pc, vengono prelevati per così dire da Amiga, e per far ciò è fornito un comando **Clock.exe** da far eseguire preferibilmente durante il boot iniziale Ms-Dos, vale a dire inserendo il comando nel file **Autoexec.bat** presente nel disco di avviamento; per chi non fosse addentro alle "segrete cose", una specie di **Startup-sequence** dei sistemi Ms - Dos. Per concludere, non resta che accennare alla compatibilità della scheda con il software dei Pc. In una parola: ottima. Naturalmente, fino all'upgrade, occorre prestare attenzione al tipo di grafica utilizzata, disponendo la Pc Board di MDA e CGA, ma in generale non si sono rilevati problemi di sorta, nel rispetto (come ovvio) della disponibilità Ram. La maggior parte dei programmi PD provati hanno funzionato senza colpo ferire, ma anche "roba" più importante come il **Word Star** ed il **Norton Editor**, svariate **calcolatrici** residenti, e persino qualche programma di comunicazione via porta seriale. "Gira" naturalmente anche il **GW Basic**, purchè, come già detto, non si adoperino risoluzioni di schermo non consentite dalla scheda grafica.

Purtroppo, per ragioni di tempo e di... organizzazione, siamo riusciti a mettere le mani solo sulla scheda XT, di cui abbiamo parlato diffusamente in queste pagine. Sappiate che proprio mentre ci state leggendo, infatti, siamo alle prese con un'altra scheda che altro non è che un **vero e proprio computer AT**, anch'essa da inserire nell'Amiga 500, di cui parleremo nel prossimo numero.

esecuzione i programmi proposti, altrimenti non funzioneranno. Si sottolinea che, per non complicare i listati, il file Magaz.txt deve esser presente nella **Root** (directory principale) del dischetto dal quale si lancia il compilatore, ma è possibile posizionarlo in qualsiasi altro **Path**. Ad esempio, nei listati in Quick Basic e Turbo Pascal riportati in queste pagine, il file Magaz.txt è presente nel Path **c:\Mariani\file**, come è possibile notare nelle istruzioni relative al colloquio con il drive. Nel listato in Turbo C, invece, abbiamo supposto che il file Magaz.txt sia presente nella Root. Per creare Magaz.txt nella directory corrente (qualunque essa sia al momento di impartire il comando Edlin), utilizziamo il programma **Edlin** del **Dos**, impartendo:

EDLIN magaz.txt

Quindi inseriamo la lista degli articoli (come vista sopra) ed alla fine dell'inserimento, con **Ctrl-c**, premiamo il tasto **E** seguito dalla pressione di Enter. Il file risulterà quindi registrato su disco e ci ritroveremo di nuovo in ambiente DOS. Per controllare la corretta digitazione del file, con...

TYPE magaz.txt

...dovremmo ottenere, su video, il file appena introdotto. In caso contrario bisognerà ridigitare il tutto. Una volta digitato e fatto partire uno dei 3 programmini pubblicati, verrà posta la seguente domanda:

Inserisci l'articolo (* Per terminare):

...e dovremo inserire il nome dell'articolo; ad esempio "Insalata" con i caratteri **minuscoli** o **maiuscoli**, a piacere. Il programma mostrerà il prezzo dell'articolo proposto e chiederà la quantità da acquistare. Su video apparirà una riga del tipo:

Articolo: Insalata ,Prezzo: 1500 ,Quantità?:

Rispondendo, ad esempio, **2** (e premendo Enter) il computer informerà sul prezzo totale della merce indicata, con la riga:

Prezzo totale: 3000

...e riproporrà la richiesta: **Inserisci l'articolo.** Digitando, ad esempio, **Pane** (e rispondendo **3** alla domanda sulla quantità) alla successiva domanda **Inserisci l'articolo** risponderemo con l'asterisco (*). Il programma terminerà visualizzando il messaggio:

Spesa totale: 13500

Cioè 10500 lire di pane (3500 x 3) + 3000 lire di insalata (1500 x 2).

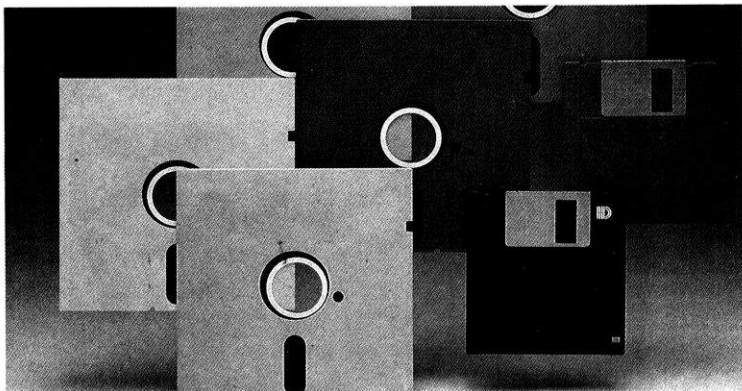
Se alla domanda "Inserisci l'articolo" rispondiamo con un nome non presente nel file "magaz.txt", il computer risponderà:

Articolo non trovato. Vuoi vedere la lista (S/N)?

Con **S**, come intuitivo, l'intera lista degli articoli (e relativi prezzi) apparirà sul video.

La banalità della procedura sviluppata consentirà di esaminare in dettaglio, e con la massima semplicità, le istruzioni da digitare nei tre linguaggi.

Alcuni comandi, per esigenze di impaginazione, **sembrano** occupare due linee nei listati pubblicati; è ovvio, invece, che vanno digitati su un'unica linea.

Al lettore, come al solito, il compito di ampliare l'archivio, magari aggiungendo campi del tipo "Unità di misura", oppure "Giacenza in magazzino", o altri ancora.

```
{ Programma Ricerca - Turbo Pascal }
Program Cerca;
```

L'istruzione uses permette di dire al compilatore che il programma usa le istruzioni contenute nella libreria specificata, in questo caso crt.

```
uses crt;
{ variabili globali }
```

Ecco la dichiarazione delle variabili, obbligatoria in Pascal, **pena segnalazione di errore** da parte del compilatore. Le **variabili, qui, sono globali,** ossia visibili dall'intero programma, **comprese procedure e funzioni.**

```
var
Totale, Prezzo, Temp : longint;
a:string[80];
q:integer;
```

Inizia qui la funzione Ricerca (di tipo **intero**), dotata di due **parametri, uno di tipo stringa** ed uno di tipo **Long**. Le istruzioni **appartenenti** alla funzione devono essere racchiuse tra **begin e end.** Nella funzione possono essere dichiarate **variabili locali, che non verranno** viste in nessun'altra sezione **del programma.**

```
{ Funzione ricerca }
function Ricerca (a:string; var Prezzo:longint):integer;
var
Trovato,k:integer;
stream:text;
Temp,Temp2:string[80];
begin
Trovato := 0;
assign(stream,'c:\mariani\file\magaz.txt');
reset(stream);
```

L'istruzione while apre un ciclo, che sarà abbandonato quando la condizione specificata tra parentesi risulterà **falsa**. Le istruzioni **appartenenti** al ciclo vanno racchiuse tra begin e end. **L'istruzione eof** diventa vera quando viene raggiunta la **fine del file indicato, e la not** la nega. In pratica, questa riga è **l'equivalente della riga basic While Not Eof(1).**

```
while (not eof(stream)) and (Trovato=0) do
begin
readln(stream,Temp);
Temp2:=copy(Temp,pos(',',Temp)+1,80);
Temp:=copy(Temp,1,pos(',',Temp)-1);
```

L'istruzione upcase trasforma in **maiuscolo** il carattere specificato. **Per trasformare l'intera stringa sarà quindi necessario ripetere l'operazione (ciclo for)** per tutti i caratteri **appartenenti ad essa.**

```
for k:=1 to length(Temp) do Temp[k]:=upcase(Temp[k]);
for k:=1 to length(a) do a[k]:=upcase(a[k]);
if a=Temp then Trovato:=1
end;
close (stream);
val(Temp2,Prezzo,k);
```

L'istruzione di assegnazione (che ha come nome della variabile il nome della funzione) produce il risultato di assegnare, a quella funzione, il valore specificato, in modo che possa essere controllato da **if Ricerca** nel main.

```
Ricerca := Trovato
end;
```

Questo è l'inizio della procedura **VediLista.** Le istruzioni appartenenti alla procedura devono essere racchiuse tra begin e end. Nella procedura possono essere dichiarate variabili locali, che quindi non verranno viste in nessun'altra sezione del programma.

```
{ Procedura vedilista }
procedure VediLista;
var
stream:text;
Temp:string[80];
begin
writeln;
assign(stream,'c:\mariani\file\magaz.txt');
reset(stream);
while not eof(stream) do
begin
readln(stream,Temp);
writeln ('Articolo, prezzo : ',Temp)
end;
close (stream); writeln
end;
```

Il **main program**, o programma primcipale, deve essere racchiuso tra **begin** e **end.** (notare il punto finale).

```
{ main program }
begin
clrscr;
writeln ('Il file di nome MAGAZ.TXT');
writeln ('deve esser presente nella Root');
Totale := 0;
clrscr;
```

Repeat apre un ciclo che deve essere chiuso con **until**. Le istruzioni tra repeat e until vengono eseguite fino a quando la condizione specificata da until risulta vera.

```
repeat
```

```
a:='';
write('Inserisci l''articolo (* per terminare) : ');
readln(a);
if a[1]<>'*' then
begin
```

L'istruzione **if** è strutturata e consente forme tipo **if.... then.... else**. In questo caso, **if Ricerca** richiama la funzione Ricerca, e si comporta a seconda del risultato da questa fornito.

```
if Ricerca(a, Prezzo)=0 then
begin
writeln; write('Articolo non trovato. vuoi vedere la
lista (S/N) : ');
readln(a);
```

Per richiamare le procedure è sufficiente specificarne il nome, seguito dagli eventuali parametri tra parentesi. In questo caso, viene richiamata la VediLista.

```
if (a[1]='s') or (a[1]='S') then VediLista
```

L'istruzione **else** è l'alternativa ad if, ossia, se la condizione specificata da if è vera, vengono eseguite tutte le istruzioni poste tra begin e end poste **dopo if**, altrimenti vengono eseguite le istruzioni comprese tra begin e end poste **dopo else**.

```
end
else begin
writeln; write('Articolo : ',a,', Prezzo : ',Prezzo,',
Quantita : ');
readln(q); if q<0 then q:=0;
writeln; writeln('Prezzo totale : ',q*Prezzo); writeln:
Totale := Totale + q * Prezzo
end;
end;
until a[1]='*';
writeln; writeln('Spesa totale: ', Totale);
repeat until keypressed;
end.
```

Con questo numero termina la fase "iniziale" dell'introduzione al **Linguaggio C**. Come i lettori avranno avuto modo di accorgersi, infatti, non ha senso interessarsi di un unico linguaggio, ma risulta decisamente più utile esaminare il modo di affrontare uno stesso problema servendosi di più linguaggi. Inizierà, dal prossimo numero (ma non è già iniziato?...) un nuovo modo di affrontare l'Informatica, quella con la A maiuscola, che non può ignorare l'evolversi dei linguaggi e delle procedure ad essi legati. I lettori più in gamba, come di consueto, avranno modo di mostrare la propria bravura accettando "sfide" che comunicheremo al più presto attraverso queste stesse pagine.

Digitando in Turbo C

Supponendo che abbiate già installato il compilatore **Turbo C** sul vostro computer, le operazioni da compiere sono le seguenti:

1) Posizionatevi nella directory che lo contiene e lanciatelo tramite il comando **TC**.

2) Premere i tasti **Alt** - **E** (edit) per accedere alla finestra di edit, e digitare con la massima cura il listato pubblicato sulla rivista.

3) Premere il tasto **F2**, inserire il nome (eventualmente completo di **Path**) del programma appena digitato per salvarlo su disco.

4) Premere i tasti **Alt** - **C** e selezionare l'opzione **Make_Exe_File** per compilare il listato.

Al termine della compilazione compare una finestra contenente alcuni dati sull'operazione compiuta. La parola **Success** indicherà una compilazione andata a buon fine, a meno che non siano presenti messaggi di **Warnings**. Bisogna infatti sottolineare che il compilatore C segnala, oltre agli errori veri e propri, anche le avvertenze (Warning), che non sono errori, ma che possono rappresentare una possibile fonte di errore, solitamente di tipo **logico**.

In quest'ultimo caso gli "inconvenienti" verranno evidenziati in una finestra in basso ed il cursore si posizionerà sul primo di questi.

Tramite i tasti freccia in alto e freccia in basso potremo spostarci lungo la lista e, premendo il tasto **F6**, posizionare il cursore sul listato sorgente nel prunto esatto in cui è stato riscontrato l'errore, che potremo controllare e correggere con facilità. Premendo di nuovo **F6** si tornerà alla finestra degli èrrori per selezionare, eventualmente, un altro errore.

Premendo **F1** nella finestra degli errori, verrà visualizzata la finestra di aiuto, relativa all'errore occorso, che contribuirà a chiarire le idee in caso di dubbi.

Bisogna ricordarsi di salvare **sempre** il listato dopo ogni correzione di errore, ed in ogni caso anche durante la digitazione per evitare di dover ridigitare il programma in caso di "inchiodamento" del computer per mancanza di tensione o altro.

Il passo 4 va ripetuto sino a che la compilazione va a buon fine: comparsa del messaggio **Success**, numero Warnings = **0** e numero Errori = **0**. Alcuni messaggi di warnings del tipo *call to function xxxxxx without prototype* possono essere prodotti dalla mancanza di righe **include**.

Il compilatore C produce il file eseguibile (suffisso **.EXE**) direttamente su disco, in modo che il programma possa girare anche in assenza del compilatore.

5) A questo punto si possono osservare i risultati premendo i tasti **Ctrl** - **F9**, sequenza che manda in esecuzione il programma appena compilato.

Se osservate elaborazioni diverse da quelle che aspettavate bisognerà stavolta controllare *non* la **sintassi** ma la **logica**, perchè sicuramente avrete sbagliato qualcosa.

Nella digitazione dei listati bisogna prestare particolare attenzione ad alcuni particolari: il compilatore differenzia le lettere **minuscole** da quelle **maiuscole**.

Il listato, comprese le variabili, va digitato esattamente come compare sulla rivista, pena segnalazioni di (in)spiegabili errori. In particolar modo, tutte le **parole chiave** andranno digitate in **minuscolo**. Inoltre, l'operatore di assegnazione eguale (=), contrariamente a quello che succede in linguaggi come il **Basic**, è diverso da quello di uguaglianza (dopio eguale, cioè ==); si dovrà, quindi, scrivere ad esempio, nel caso di assegnazione (**A = B**) e, in caso di confronto...

if (A==B)

...e non...

if (A=B)

...che in C equivarrebbe ad una istruzione di assegnazione **A = B** con la conseguenza che verrebbe effettuato un confronto del risultato con zero, del tipo...

if (A < > 0)

Tale svista, naturalmente, produce un risultato completamente diverso da ciò che si aspetta, pur se viene segnalato dal compilatore con un Warning del tipo *Possibly incorrect assignment* che, però, non arresta la compilazione... In C l'operatore **diverso da** è composto da un punto esclamativo ed un uguale (**!=**) e **non**, come in altri linguaggi, dai segni minore e maggiore (**< >**). L'espressione **se A è diverso da B** diventa, in C:

if (A != B)

Le stringhe sono trattate TUTTE come puntatori: non si può assegnare una stringa ad un'altra con un'istruzione del tipo...

Stringa2 = Stringa1

Quest'ultima forma sintattica, in C, significherebbe: **il puntatore Stringa2 punta ora a Stringa1**.

Si otterrebbe, come risultato, non solo il fatto che Stringa1 e Stringa2 punterebbero alla stessa area di memoria (cioè risulterebbero una **variabile unica**), ma anche che il puntatore all'area precedentemente puntata da Stringa2 verrebbe **perso**, non permettendo di recuperare in alcun modo quell'area di memoria.

L'assegnazione tra le stringhe deve invece essere effettuata tramite l'istruzione...

strcpy (Stringa2, Stringa1)

...che copia le due stringhe correttamente, lasciando inalterati i relativi puntatori.

In ogni caso il Turbo C possiede un **help** in linea, disponibile in ogni momento con la pressione del tasto **F1**, grazie al quale si accede all'help generale, che contiene informazioni relative all'editor ed al pacchetto. Premendo **Ctrl** - **F1**, con il cursore posizionato su di una parola chiave, verrà visualizzato l'help ad essa relativo corredato, a volte, di un esempio di programmazione.

Se viene premuto **Ctrl** - **F1** quando il cursore non è posizionato su di una parola chiave, viene proposta la lista di tutte le parole e si potrà scegliere quella della quale si vuole l'help.

Digitando in Turbo Pascal

Si suppone che abbiate già installato il compilatore sul vostro computer, seguendo le istruzioni riportate nei manuali. Per digitare, e compilare, i programmi riportati sulle nostre pagine, è necessario attenersi scrupolosamente alle seguenti operazioni, che si riferiscono alla versione **5.5** del compilatore originale **Borland**:

1) Richiamare il compilatore tramite il comando **Turbo**.

2) Con i tasti **Alt** - **E** accedere alla finestra di edit, e trascrivere scrupolosamente il listato che interessa.

3) Terminata la digitazione è consigliabile salvare subito il programma: premere il tasto **F2** per inserirne il nome (completo di eventuale Path) e salvarlo su disco.

4) Con i tasti **Alt** - **F9** si compila il listato. Dopo un tempo (comunque breve) dipendente dalla lunghezza del listato e dalla velocità del computer usato, il compilatore visualizzerà una finestra riportante alcuni dati sulla compilazione (**Success** compare in caso di compilazione andata a buon fine). Nel caso fossero segnalati errori, questi verranno evidenziati in **rosso** (in alto sullo schermo), ed il cursore si posizionerà sul primo di questi, permettendo all'operatore di controllare e correggere l'errore. Con la pressione di **F1** verrà automaticamente aperta una finestra, ove sarà possibile informarsi sull'errore occorso.

E' bene salvare il listato dopo aver apportato anche la minima correzione. Ciò evita di digitare nuovamente il programma in caso di "inchiodamento" del computer per mancanza di tensione o a causa di errori terrificanti. Il passo 4 va ripetuto sino a che la compilazione va a buon fine.

5) A questo punto si possono osservare i risultati premendo i tasti **Ctrl** - **F9**, che mandano in esecuzione il programma compilato. Nel caso non si verifichi ciò che ci si aspettava, bisognerà controllare non la sintassi, ma la logica del listato.

Nella digitazione dei listati in Turbo Pascal bisogna prestare attenzione ad alcuni particolari:

Tutte le istruzioni (tranne i commenti e quelle che precedono istruzioni end) devono terminare con il carattere di punto e virgola (;).

L'operatore di **assegnazione** non è il semplice segno eguale (=), ma la coppia di caratteri doppio punto e uguale (:=).

Solo il **confronto** viene effettuato con il carattere eguale (=). Sono quindi istruzioni corrette "**a := 5**" e "**if (a=0) then...**", ma non "**a = 5**".

Il compilatore offre un help in linea, disponibile in ogni momento premendo il tasto **F1**, che consente l'accesso all'help generale. Questo contiene informazioni relative sia all'editor che al pacchetto in generale. Premendo **F1** insieme a **Ctrl** (con il cursore posizionato su di una parola chiave), verrà visualizzato l'help relativo alla parola stessa fornendo, talvolta, esempi di programmazione.

Premendo **Ctrl** - **F1** quando il cursore non è posizionato su di una parola chiave, viene proposta la lista di tutte le parole disponibili e si potrà scegliere quella della quale si vuole l'help.

Il T.Pascal non produce files con suffisso **Exe** direttamente su disco, a meno che non lo si specifichi tramite l'opzione presente nella finestra **Options**, accessibile tramite i tasti **Alt** - **O**.

Originale è meglio

I linguaggi moderni, come il **Turbo Pascal**, il **Turbo C** ed il **Quick Basic**, presentano somiglianze incredibili, tanto che un listato, digitato in uno dei tre linguaggi, può essere adattato con grande facilità alla sintassi richiesta dagli altri due.

Molti nostri lettori forse posseggono **copie** di favore (l'eufemismo è d'obbligo) dei potenti compilatori ma, ovviamente, non riescono ad usarli correttamente a causa della mancanza del manuale o per l'incapacità di tradurre, dall'inglese, i messaggi eventualmente visualizzati, nel caso si possegga una versione americana del prodotto.

Qualcuno, forse, si sarà procurato un volume con l'intenzione di supplire alla mancanza dei manuali, ma si sarà accorto che i libri posti in vendita sono (troppo) spesso una **brutta** copia (peraltro **incompleta**, per ovvie ragioni di copyright) dei manuali inclusi nelle confezioni originali dei linguaggi.

Lo scopo delle note, presenti in queste pagine, relative ad un uso sommario (ma corretto) dei linguaggi eventualmente piratati, non rappresentano affatto un invito alla diffusione della pirateria. Servono, al contrario, per dimostrare che i più noti linguaggi oggi disponibili offrono una sintassi abbastanza simile a quella del linguaggio tradizionalmente più diffuso (il **Basic**) ed un implicito invito alla digitazione dei listati ed alla comprensione delle enormi potenzialità offerte dai pacchetti originali.

La conclusione, pertanto, è solo una: dopo essersi resi conto che imparare un nuovo linguaggio è molto più semplice di quanto si sospettava, il passo successivo è, "ovviamente", quello di procurarsi la **confezione originale**, magari in italiano, rinunciando ad usare la copia piratata in proprio possesso e decidendo, finalmente, di avere a disposizione un pacchetto **completo**, fornito di informazioni **precise** e certamente più **ufficiali** di quelle offerte, a prezzo relativamente caro, da volumi alternativi che si limitano a parlare genericamente di un certo linguaggio.

Il limitato prezzo al pubblico, e la possibilità di ottenere facilitazioni con le software house che producono i pacchetti, non faranno certo rimpiangere il denaro speso per l'acquisto della confezione originale.

Digitando in QBasic

Supponendo che abbiate già installato il compilatore sul computer, seguendo le istruzioni riportate nei manuali, le operazioni da compiere (riferite alla versione **4.5** della **Microsoft**) sono le seguenti:

1) Attivare il compilatore con **QB**.

2) Trascrivere con la massima attenzione il listato che interessa.

3) Terminata la digitazione, premere i tasti **Alt** - **F** e, subito dopo, il tasto **S** per salvare su disco il programma indicandone il nome (completo di eventuale **Path**).

4) Premere i tasti **Shift** - **F5** per compilare (ed eseguire) il listato appena digitato. Dopo un certo tempo il programma andrà in esecuzione. In caso contrario la compilazione si arresta, viene visualizzata la finestra con la descrizione dell'errore riscontrato ed il cursore si posiziona su di esso, permettendone la correzione.

E' bene salvare sempre il listato dopo ogni correzione o modifica e, magari, anche durante la digitazione. Ciò eviterà di digitare nuovamente il programma in caso di "inchiodamento" del computer per mancanza di tensione o altro. Il passo **4** va ripetuto sino a che la compilazione va a buon fine.

Il compilatore QuickBasic non produce il file eseguibile (suffisso **.EXE**) direttamente su disco. Per ottenerlo, quindi, bisogna, nell'ordine:

Premere i tasti **Alt** - **R**, quindi **X** (Make e**X**e file). All'apparire della finestra di compilazione, premere i tasti **Alt** - **A** (Produce stand - **A**lone file), quindi **Alt** - **M** (**M**ake exe), ed attendere che il processo di compilazione abbia termine. "Uscendo" dall'ambiente QuickBasic (tramite **Alt** - **F** e poi **X**) si potrà far girare il programma senza che sia presente il compilatore.

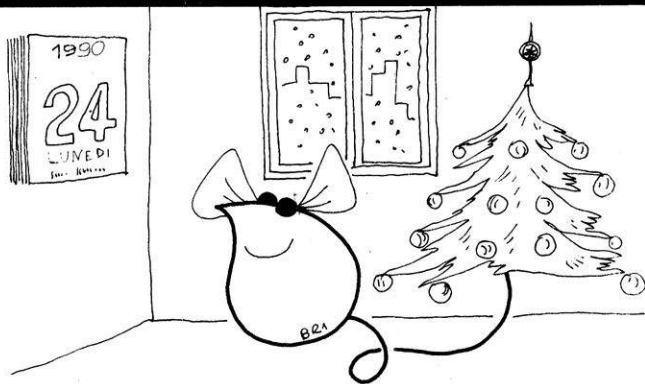
5) Se, osservando i risultati prodotti dal programma, si devessero notare elaborazioni diverse da quelle che ci si aspettava, bisognerà controllare non la sintassi, ma la logica del programma.

Il compilatore offre un controllo di sintassi "in linea", ossia sempre attivo, che segnala, tramite finestre su video, errori commessi addirittura in fase di digitazione.

I numeri di linea non sono necessari (a differenza del predecessore GWbasic). Le eventuali istruzioni **GoTo**, **GoSub**, che si riferiscono a linee di programma individuabili dal numero di linea, potranno essere usate etichette alfanumeriche, lunghe fino a 40 caratteri e seguite dal carattere doppio punto (:). Tutte le parole chiave digitate saranno automaticamente convertite in caratteri **maiuscoli**, mentre le variabili saranno lasciate come digitate dall'operatore, tenendo però presente che il compilatore non differenzia i caratteri minuscoli da quelli maiuscoli: la variabile **Pippo** viene considerata uguale alla variabile **piPPo**, oppure **pippo**.

Le **procedure** (SUBs) e **funzioni** (FUNCTIONs) **non** vanno digitate nel testo del programma. Per inserirle deve essere selezionata l'opzione **New Sub** (oppure **New Function**), tramite il menu **Edit**, selezionabile con i tasti **Alt** - **E**. Premendo il tasto **F2** si avrà un elenco delle funzioni e procedure comprese nel main program, per selezionarne una tramite i tasti cursore e Enter.

Il compilatore QuickBasic offre un help in linea, disponibile in ogni momento con la pressione del tasto **F1**. Premendolo si accede all'help generale, che contiene informazioni relative sia all'editor sia al pacchetto in generale.

Premendo F1 insieme a **Shift** (con il cursore posizionato su di una parola chiave), verrà visualizzato l'help ad essa relativo corredato, a volte, di esempi di programmazione. Se viene premuto Shift - F1 quando il cursore non è posizionato su di una parola chiave, viene proposto l'help generale.

di Davide Currò

MESSAGGI CIFRATI DA UN'AMIGA

Un'originale tecnica di crittografia viene sviluppata in Amigabasic

Il programma trattato in queste pagine permette di creare messaggi in codice, di stamparli, salvarli su disco e infine di decodificarli, il tutto in modo agevole.

La tecnica usata riproduce un sistema chiamata **Codice Siepi** che consiste nel nascondere il messaggio da rendere segreto all'interno di una griglia rettangolare. Il sistema scompone il messaggio nelle singole lettere dalle quali è formato e le inserisce nella griglia destinata ad accoglierle, seguendo un preciso schema geometrico.

Nella figura 1 lo schema in questione è rappresentato dalle cinque "X" che supporremo disposte a casaccio dall'utente. La disposizione delle "X", ripetute ciclicamente nella griglia stessa, costituisce il vero e proprio "codice" che possiamo rintracciare, ripetuto tre volte, nella figura 2.
Supponiamo, appunto, che il messaggio da codificare sia **Buongiorno amici**.

Una volta inserite nella griglia le lettere che formano il messaggio, alcuni spazi rimangono vuoti (il codice-schema geometrico, infatti, non utilizza quasi mai tutte le caselle) e vengono riempiti con lettere casuali (vedi figura 3), in modo che, alla fine delle operazioni, la griglia rappresenti un'accozzaglia di lettere senza senso. Così facendo, il messaggio nascosto può essere compreso solo da chi conosce l'ordine in cui vanno letti i caratteri per ricostruire la sequenza del testo originario.

Fig. 1: Esempio di griglia descritto nell'articolo

Il programma

Vediamo ora come, il nostro programma, renda automatica la procedura descritta sommariamente. Da sottolineare, anzitutto, che per un suo corretto funzionamento è necessario predisporre, da **Preferences**, la modalità **60 colonne**; in caso contrario possono verificarsi disallineamenti del testo. Appena digitiamo Run ci troviamo di fronte al menu principale che presenta 6 opzioni:

1 crea messaggio;
2 decifra messaggio;
3 stampa messaggio;
4 carica messaggio;
5 save messaggio;
6 fine.

Premendo il numero corrispondente all'opzione desiderata verrà visualizzato un sottomenu che presenta altre operazioni eseguibili. Analizziamo singolarmente ogni opzione. Selezionando la prima (**crea messaggio**), entreremo nel sottomenu che permette di creare un messaggio in codice. Per fare ciò dobbiamo inserire gli elementi necessari per la codificazione del messaggio stesso, selezionando le singole opzioni del sottomenu: esaminiamole.

La prima chiede le **dimensioni** della griglia in cui andrà inserito il messaggio al momento della codificazione (vedi oltre, opzione 7).

Le dimensioni massime della griglia sono di 30 caratteri orizzontalmente e di 32 verticalmente. Pertanto valori superiori, nulli o negativi non verranno accettati. Per seguire l'esempio delle figure, realizzerete una griglia di **7 righe** (= altezza caselle) e **9 colonne** (= larghezza). L'opzione 2 calcola le dimensioni della griglia più appropriate, permettendo di risparmiare tempo nella ricerca empirica di queste, attraverso le opzioni 1 e 6 (descritte in seguito). Spesso, infatti, la griglia risulta troppo piccola per

B		I		A				
	U		O		M			
	O		R		I			
	N		N		C			
G		O		I				

Figura 2: Codificazione del messaggio "Buongiorno amici"

contenere interamente il messaggio, oppure esageratamente grande, costringendoci a ritoccare più volte le dimensioni con l'opzione 1. La procedura ricerca automaticamente le dimensioni più adatte, eseguendo calcoli basati sulla lunghezza del messaggio e sul tipo di codice.

Per chi inizia

Il programma di queste pagine è destinato a chi si è procurato da poco tempo un Amiga. La sua lunghezza, infatti, è sufficientemente breve(!) da invogliare il lettore alla sua digitazione. Questa deve essere effettuata prestando la massima attenzione ai valori numerici presenti al suo interno; con particolare cura devono esser trascritti gli argomenti delle istruzioni.

Anche se non riuscite a ben comprendere il reale significato delle varie istruzioni, non scoraggiatevi: digitate il tutto e registrate su dischetto (comando **Save**) il programma pubblicato.

Il valore del programma è anche didattico. E' infatti possibile rendersi conto delle varie difficoltà che si incontrano nello scrivere un listato, benchè semplice come quello che compare in queste pagine. Figuratevi, quindi, la pazienza che è necessaria per realizzare listati pieni di sprite, schermate in alta risoluzione, musica ed animazioni varie!

A volte può capitare che il programma presenti malfunzionamenti a causa di una non corretta trascrizione del listato.

Nei casi più fortunati viene emessa una segnalazione di Syntax Error oppure di Illegal Quantity Error. In altri casi si rischia di bloccare il computer. C'è infine una terza eventualità: in caso di errore il programma gira e non segnala errori, ma l'elaborazione procede in maniera anomala ed erronea.

Si consiglia ancora, pertanto, di digitare il programma con la massima attenzione e di registrarlo sempre, prima di impartire il comando Run.

Da quanto detto si capisce che non è possibile selezionare l'opzione senza aver prima inserito sia il messaggio sia il codice (utilizzando le opzioni 3, 4 e 5).

Talvolta le dimensioni consigliate possono risultare insufficienti per accogliere l'intero messaggio (ciò è dovuto all'approssimazione dei valori nel calcolo); in tal caso bisognerà ingrandire la griglia con l'opzione 1.

L'opzione **3** serve per inserire, graficamente, il codice: questo è formato da una serie di caselle (non più di 20) disposte a formare, come già detto, uno schema geometrico. Riferendosi ancora alla figura 1, il codice è costituito dall'area in grassetto; tale cornice è stata tracciata (pur non comparendo nella videata) per sottolineare che il programma assume, come codice, l'area del rettangolo posizionato nell'angolo superiore sinistro della griglia, delimitato dalla "X" che si trova più in basso e dalla "X" che si trova più a destra.

Ed è proprio questo, il "rettangolo" considerato dal nostro codice, non l'intera griglia che appare sul video (la quale potrebbe anche variare di dimensioni) e nemmeno le sole X. Nel comporre il codice dovremo quindi sempre posizionare lo schema in alto a sinistra nella griglia che appare sullo schermo.

Con l'opzione **7** lo schema (ovvero il rettangolo "ideale") viene riprodotto più volte nella griglia, partendo dall'angolo in alto a sinistra e procedendo orizzontalmente. Quando non c'è più spazio per un altro schema, cioè si è arrivati al limite destro della griglia, il programma va "a capo" e continua ad affiancare gli schemi, orizzontalmente, nelle caselle sottostanti.

All'interno delle caselle che costituiscono gli schemi disposti nella griglia (quelle contrassegnate dalle X) vengono poi inserite le lettere che costituiscono il messaggio originario (procedendo sempre da sinistra verso destra e dall'alto verso il basso).

L'esempio di figura 2, già citata, mostra tre schemi affiancati, in cui, al posto delle X, sono state sostituite le lettere del messaggio "Buongiorno amici".

A questo punto rimangono delle caselle bianche, non riempite dal messaggio; verranno riempite con caratteri casuali. Il tutto apparirà come una griglia di lettere disposte in modo apparentemente insensato, ma che contengono un messaggio ben preciso che può essere compreso solo da chi, conoscendo il codice,

B	A	I	S	A	F	C	G	M
Q	U	M	O	B	M	S	V	S
K	O	R	R	X	I	B	Q	H
A	N	A	N	L	C	C	E	W
G	M	O	A	I	Z	R	B	N
K	A	Y	K	K	S	E	H	L
A	H	E	I	Q	L	E	W	M

Fig. 3: Il messaggio "Buongiorno amici" non è più riconoscibile

è in grado di individuare i caratteri ripercorrendo lo schema originario.

La figura 3 mostra un possibile stadio "finale" del messaggio di figura 2; gli spazi bianchi della griglia sono stati colmati con lettere casuali ed il rettangolo è totalmente incomprensibile.

L'immissione del messaggio nella griglia avviene, come già detto, solo con l'opzione 7; prima, con l'opzione 3, dobbiamo solo creare lo schema - codice posizionando la freccia del mouse **al**

livelli di **compenetrazione** del codice. E' possibile creare particolari schemi in modo che le loro aree possano venir sovrapposte parzialmente, permettendo di complicare ulteriormente l'operazione di codifica o di risparmiare spazio all'interno della griglia. La struttura degli schemi (vedi la figura 4) deve essere tale che questi possano venir sovrapposti di una o più caselle, senza però che le caselle contenenti le X, cioè quelle dove verranno disposte le lettere del messag-

Copialo per telefono

Anche i listati presenti in queste pagine può esser tirati giù per mezzo del modem; se, ovviamente, ne possedete uno.
La procedura per collegarsi con la nostra banca dati (attiva 24 ore su 24) è riportata su altra parte della rivista.

centro della casella che si vuole selezionare e premendo il tasto sinistro. Nella casella comparirà una "X" per segnalare che ora quella casella fa parte dello schema - codice. Le X possono essere posizionate come più vi aggrada, formando le più strane combinazioni: l'importante è di **non scrivere più di una sola X** per ciascuna casella, cosa che genererebbe la totale distruzione del messaggio al momento della codificazione.

Dopo aver selezionato le caselle desiderate, posizionate la freccia sulla casella "**Fine**" e cliccate una voltà (sempre con il tasto sinistro): sullo schermo comparirà il numero di caratteri che compongono lo schema; premete quindi un tasto per finire. Un'ultima avvertenza: prima di selezionare l'opzione 3, dovete necessariamente immettere le dimensioni della griglia (con 1): non è necessario che siano le dimensioni definitive (queste potranno venir modificate in seguito), ma devono essere sufficientemente grandi da contenere lo schema del codice.La quarta opzione permette di selezionare i

gio, si sovrappongano le une sulle altre. In questo caso, infatti, il messaggio non sarebbe più conprensibile poichè le prime sarebbero coperte dalle seconde. In pratica, durante la sovrapposizione, le X del secondo schema prendono il posto delle caselle vuote all'interno dell'area del primo schema (nello stesso modo in cui il pezzo di un puzzle si sovrappone a un altro pezzo, andando a colmare lo spazio vuoto all'interno di quest'ultimo).

La figura 4 mostra come sono state sovrapposte le aree degli schemi di figura 2 assegnando il valore 1 al livello di compenetrazione orizzontale. E' evidente il risparmio di spazio che comporta. Dopo aver composto il codice con l'opzione 3, potremo assegnare i valori desiderati ai livelli.. Il valore 0 è quello per cui non si ha nessuna sovrapposizione; dando come valore 1 gli schemi verranno sovrapposti di una casella; con 2, di 2 caselle e così via. Se, invece, diamo valori negativi, cioè -1, -2 etc, gli schemi verranno distanziati di 1, 2... caselle. La sovrapposizione può avvenire **orizzontalmente** (livello orizzontale), sovrapponendo lateralmente più schemi, **verticalmente** (livello verticale), sovrapponendo la fila inferiore a quella superiore, o contemporaneamente in entrambi i modi. Attenti, però, a non esagerare per evitare sovrapposizioni che cancellerebbero parte del messaggio.

La quinta opzione permette di inserire il messaggio da codificare.

Questo non deve contenere spazi bianchi poichè verrebbero trattati come caratteri (e creerebbero confusione nel momento della codificazione), nè virgole, che comprometterebbero un corretto salvataggio del messaggio sul dischetto.

Prima di iniziare l'inserimento, viene richiamata automaticamente l'opzione **6** che fornisce la lunghezza massima che può avere il messaggio in relazione al codice e alle dimensioni della griglia. Non è necessario rispettare il valore che viene dato, a patto che poi ci si ricordi di modificare la griglia con l'opzione 2.

L'opzione 6, come già detto, calcola la lunghezza massima del messaggio da codificare. La **7**, invece, procede alla compilazione del messaggio.

Il metodo usato è già stato descritto in precedenza; se tuttavia la griglia ha un'altezza maggiore di 8 caselle, questa viene visualizzata sul video in più riprese, per motivi di spazio. Per comprendere meglio come avviene la compilazio-

B	I	A						
	U	O	M					
	O	R	I					
	N	N	C					
G	O	I		I				

Fig. 4: Livelli di compenetrazione.. Si noti il riavvicinamento dei tre gruppi di caratteri che, nella figura 1, risultavano distanziati.

ne, fermate in vari istanti l'andamento del programma durante l'elaborazione, agendo sul tasto destro del mouse.

E' intuibile che, per creare un messaggio, è obbligatorio passare per i punti **1, 3, 5, 7,** mentre gli altri sono facoltativi. Inoltre è importante notare che, una volta inseriti i dati, questi rimangono memorizzati: è possibile quindi creare un messaggio, salvarlo su disco e procedere alla creazione di un nuovo messaggio utilizzando solo il punto 5 (inserisci messaggio) senza esser costretti ad inserire nuovamente il codice, i livelli di compenetrazione e tutto il resto (ovviamente così facendo il secondo messaggio verrà compilato usando la stessa procedura del primo); oppure si possono salvare messaggi uguali modificando solo il codice.

Con 8 si ritorna al menu principale.

Decodifica

Selezioniamo ora la seconda opzione, cioè **decifra messaggio**. Si presenta un nuovo sottomenu con 6 opzioni: i punti 1, 2, 3 e 6 li conosciamo già; il punto **4** chiede di inserire il messaggio da tradurre dalla tastiera (dopo aver immesso le dimensioni della griglia da cui è composto). Questa opzione è utile per chi volesse tradurre un messaggio pervenuto su un foglio di carta: dapprima, purtroppo, si riscrive l'intera griglia; poi si procede a decodificarla.

Per la decodificazione bisogna selezionare il punto 5, dopo aver fornito al programma gli elementi necessari (messaggio, codice, dimensioni, etc). In realtà, come già detto, non è necessario inserire ogni volta tutti i dati, se questi sono già presenti "in memoria". Per convincervene create un messaggio (attraverso l'opzione **crea messaggio**) e poi, senza interrompere il programma, entrate in questo sottomenu (**decifra messaggio**) e selezionate il punto **5**: verrà decodificato il messaggio presente in memoria, cioè quello che avete appena creato. Da ciò si capisce che, per decifrare un messaggio, potete anche caricarlo da disco e poi decifrarlo direttamente (aggiungendo il codice nel caso in cui il messaggio sia stato salvato senza codice).

Tornate ora al menu principale e selezionate l'opzione **3**: con la prima opzione potrete stampare un messaggio in codice (cioè la griglia di lettere incomprensibili) dopo averlo creato in precedenza oppure dopo averlo caricato da disco. Con il punto 2, invece, potrete stampare la traduzione di un messaggio in codice, ma solo dopo aver decifrato il messaggio con le procedure sopraelencate.

La **quarta** opzione del menu principale permette di caricare un file da disco: viene richiesto il nome del file e il drive sorgente, dopodichè il file viene caricato e il programma ci informa se il file è completo di codice, o meno.

Se il codice non è presente, non è possibile decifrare il messaggio contenuto nel file, a meno che non lo si inserisca a parte da tastiera.

L'opzione 5 del menu principale provvede, diversamente, a salvare un file su disco: anche qui viene richiesto il nome da assegnare al file e il drive destinazione. Inoltre bisogna scegliere il tipo di salvataggio; cioè completo di codice o senza. Un utile suggerimento è quello di salvare tutti i file "segreti" senza codice su un disco, e salvare un file con il codice traduttore (lo stesso degli altri) su un disco diverso (ad esempio nelle directory di sistema di un word processor, magari con un nome inglese per camuffarlo meglio). Ogni volta che vorrete tradurre un messaggio precentemente memorizzato, o crearne uno nuovo, caricherete prima il file-messaggio completo di codice (quello che avete nascosto); in questo modo il codice che siete soliti usare sarà già bell'e pronto in memoria senza che lo dobbiate inserire da tastiera.

Per tradurre un messaggio salvato su disco senza codice (è ovvio che in questo caso il codice deve essere uno solo e lo stesso per tutti i messaggi) non farete altro che caricarlo e poi tradurlo. Durante il caricamento del file da tradurre, infatti, il messaggio proveniente dal dischetto si sovrappone a quello in memoria, ma il codice attualmente in memoria viene conservato tale e quale.

Una volta presente in memoria il file-codice, basterà sovrapporre il nuovo messaggio su quello già presente (il quale non è altro che un messaggio inutile che serve soltanto a salvare su disco il codice segreto), e poi eseguire il punto 7 del sottomenu **crea messaggio**, dal momento che tutti gli altri elementi sono già presenti in memoria da quando avete caricato il file-codice.

Al limite, potrebbe rendersi necessario adattare le dimensioni della griglia alla lunghezza del nuovo codice.

Potrete crearvi, insomma, un *libro nero* che potrà leggere solo chi sarà in possesso del file-codice (cioè voi).

Il programma è scritto interamente in **Amigabasic** e non fa uso di tecniche particolari come librerie e cose simili: pertanto è molto semplice modificare e perfezionare le procedure del programma. Ad esempio, non dovrebbe esser difficile realizzare una procedura che prelevi il contenuto di un file creato con un word-processor e lo trasformi in un messaggio in codice, procedendo poi a salvarlo su disco nel nuovo "formato". Magari provvedendo a spezzarlo in più files qualora il testo sorgente fosse troppo lungo per essere inserito in una sola griglia.

```
' CODICE SIEPI
' by Davide Curro' 1990
PRINT "Attenzione: settare PREFERENCES a 60 caratteri!"
PRINT "(premi un tasto) "
attesatasto:
A$ = INKEY$: IF UCASE$ (A$) = ""THEN GOTO attesatasto:
main:
RANDOMIZE TIMER: DIM cod (21, 2), scrit$ (32, 30), p% (1530)
xc = 20: yc = 18: xp = 5: yp = 4: q$ = CHR$ (13) : WIDTH 60

opzioni:
CLS: LINE (43, 13) - (260, 30), , b
LOCATE 3, 6: PRINT "CODICE SIEPI - MENU'": PRINT
PRINT " 1 = CREA MESSAGGIO"q$" 2 = DECIFRA MESSAGGIO"
PRINT " 3 = STAMPA MESSAGGIO"q$" 4 = CARICA MESSAGGIO"
PRINT " 5 = SAVE SU DISCO"q$" 6 = FINE"
LINE (6, 5) - (420, 120), , b
GOSUB tasto1: LOCATE 15, 2: b = VAL (A$) : IF b = 6 THEN END
ON b GOSUB fames, traduce, stampante, drive, drive
GOTO opzioni

stampante:
CLS: LOCATE 3, 1
PRINT "1 Stampa messaggio in codice"
```

```
PRINT "2 Stampa messaggio tradotto"
PRINT "3 Torna al menu' principale"
GOSUB tasto1: b = VAL (A$)
IF b = 3 THEN RETURN
PRINT "Posiziona la stampante": GOSUB tasto
ON b GOSUB stammes, stamsol: GOTO stampante
stammes:
IF scrit$ (1, 1) = ""THEN
LOCATE 19, 2: PRINT "Nessun messaggio da stampare"
GOSUB tasto: RETURN
END IF
FOR i = 1 TO al: FOR i1 = 1 TO lg
LPRINT scrit$ (i, i1) ; : NEXT: LPRINT: NEXT: RETURN
stamsol:
WIDTH LPRINT 75: LPRINT me$: RETURN
fames:
CLS: LOCATE 3, 1
PRINT " 1 dimensioni della GRIGLIA"
PRINT " 2 calcola dimensioni appropriate"
PRINT " 3 inserisci CODICE"
PRINT " 4 livelli di compenetrazione"
PRINT " 5 inserisci MESSAGGIO"
PRINT " 6 numero di lettere utilizzabili"
PRINT " 7 COMPILA messaggio in codice"
PRINT " 8 torna al menu' principale"
GOSUB tasto1: b = VAL (A$) : IF b = 8 THEN RETURN
ON b GOSUB hl, cdim, code, livc, cmes, carat, comes
GOTO fames
cdim: ' **** calcola dimensioni migliori ****
IF al = 0 OR cod (0, 1) = 0 OR me$ = ""THEN RETURN
ix = 0: iy = 0: i3 = LEN (me$)
caldim2:
ix = ix + 1: IF ix > 30 THEN ix = 30
lg = ix: GOSUB calcolacar
IF car > = i3 AND (car - i3) = < 20 THEN GOTO caldim3
iy = iy + 1: al = iy: GOSUB calcolacar
IF car > = i3 AND (car - i3) = < 20 THEN GOTO caldim3
IF ix = 30 AND iy = 32 THEN
PRINT " Messaggio troppo lungo per l'intera griglia"
GOSUB tasto: RETURN
END IF: GOTO caldim2
caldim3:
al = iy: lg = ix: GOSUB carat: LOCATE 19, 2
PRINT SPACE$ (10) : LOCATE 4, 1
PRINT " Valori consigliati: "q$" altezza = "; al
PRINT " larghezza = "; lg: GOSUB tasto: RETURN

carat: ' *** calcola caratteri utilizzabili ***
CLS: IF al = 0 OR cod (0, 1) = 0 THEN RETURN
GOSUB calcolacar
PRINT "Si consiglia di non superare i "; car; "caratteri"
GOSUB tasto: RETURN

cmes: CLS ' **** Chiede messaggio da codificare *****
PRINT "Continui il messaggio in memoria ? (S) "
GOSUB tasto: IF A$ < > "s"THEN me$ = ""
GOSUB carat: LOCATE 3, 2: PRINT "Inserire il ";
PRINT "messaggio senza spazi o virgole DEL = corregge"
WIDTH 60: LOCATE 6, 1: PRINT me$;
scritta:
A$ = "": WHILE A$ = "": A$ = INKEY$ : WEND
IF A$ = CHR$ (127) THEN
me$ = LEFT$ (me$, LEN (me$) - 1)
psx = POS (x) - 1: psy = CSRLIN
IF psx < 1 THEN psx = 60: psy = psy - 1
LOCATE psy, psx
PRINT "": LOCATE psy, psx: GOTO salta3
END IF
IF A$ = CHR$ (13) THEN RETURN
me$ = me$ + UCASE$ (A$) : PRINT UCASE$ (A$) ;
salta3: psx = POS (x) : psy = CSRLIN: LOCATE 4, 1: PRINT LEN
(me$) ;
PRINT "caratteri": LOCATE psy, psx: GOTO scritta

comes:
IF me$ = ""OR al = 0 OR cod (0, 1) = 0 THEN RETURN
FOR i = 1 TO al: FOR i1 = 1 TO lg: scrit$ (i, i1) = ""
NEXT: NEXT: GOSUB griglia: md$ = me$: liv = - 1: lio = - 1: sc
= 0
livelv: liv = liv + 1: lio = - 1
livelo: lio = lio + 1
FOR i = 1 TO nc
lx = cod (i, 1) + cod (0, 1) * lio
IF lio > 0 THEN lx = lx - compo * 2 * lio
ly = cod (i, 2) + cod (0, 2) * liv
IF liv > 0 THEN ly = ly - compv * 2 * liv
lysc = ly - sc * 16
IF LEN (md$) = 0 THEN GOTO coprebianco
IF lx > lg * 2 + 1 THEN GOTO torna
IF ly > al * 2 + 1 THEN GOTO torna
IF lysc > 16 OR lysc < 2 THEN GOTO salta
LOCATE lysc, lx: PRINT LEFT$ (md$, 1)
salta: scrit$ (ly / 2, lx / 2) = LEFT$ (md$, 1)
md$ = RIGHT$ (md$, LEN (md$) - 1)
torna: NEXT
IF lysc > 16 THEN sc = sc + 1: GOSUB griglia
IF lx > lg * 2 + 1 THEN GOTO livelv
IF ly > al * 2 + 1 THEN
LOCATE 19, 2: PRINT "messaggio non intero"
GOSUB tasto1: GOTO coprebianco
END IF: GOTO livelo
coprebianco:
sc = 0: GOSUB griglia: FOR i = 1 TO al: FOR i1 = 1 TO lg
IF scrit$ (i, i1) = ""THEN scrit$ (i, i1) = CHR$ (RND * 25 + 65)
lysc = i * 2 - sc * 16
IF lysc > 16 OR lysc < 2 THEN GOTO salta1
LOCATE lysc, i1 * 2: PRINT scrit$ (i, i1)
salta1: NEXT
IF lysc = 16 THEN sc = sc + 1: GOSUB tasto: GOSUB griglia
NEXT: GOSUB tasto: RETURN

traduce:
CLS: LOCATE 3, 1
PRINT " 1 dimensioni GRIGLIA"
PRINT " 2 inserisci CODICE traduttore"
PRINT " 3 livelli di compenetrazione"
PRINT " 4 inserici MESSAGGIO da tradurre"
PRINT " 5 TRADUCE messaggio in memoria"
PRINT " 6 torna a menu' principale"
GOSUB tasto1: IF VAL (A$) = 6 THEN RETURN
ON VAL (A$) GOSUB hl, code, livc, insemes, deciframes
GOTO traduce

insemes:
sc = 0: GOSUB griglia: LOCATE 18, 2: PRINT "Scrivi il ";
PRINT "messaggio linea per linea (DEL = corregge) "
FOR i = 1 TO al: FOR i1 = 1 TO lg
A$ = "": WHILE A$ = "": A$ = INKEY$: WEND
scrit$ (i, i1) = UCASE$ (A$) : lysc = i * 2 - sc * 16
IF A$ = CHR$ (127) THEN
i1 = i1 - 2: IF i1 < 0 THEN i1 = lg - 1: i = i - 1: lysc = i * 2 - sc * 16
LOCATE lysc, i1 * 2 + 2: PRINT "x": GOTO salta4
```

```
END IF: LOCATE lysc, i1 * 2: PRINT UCASE$ (A$)
salta4: NEXT
IF lysc = 16 THEN sc = sc + 1: GOSUB griglia
NEXT: GOSUB tasto: RETURN
deciframes:
IF scrit$ (1, 1) = ""OR al = 0 OR cod (0, 1) = 0 THEN RETURN
sc = 0: me$ = "": n = 1: GOSUB griglia: GOSUB compila
liv = - 1: lio = - 1
livelv1: liv = liv + 1: lio = - 1
livelo1: lio = lio + 1
FOR i = 1 TO nc
lx = cod (i, 1) + cod (0, 1) * lio
IF lio > 0 THEN lx = lx - compo * 2 * lio
ly = cod (i, 2) + cod (0, 2) * liv
IF liv > 0 THEN ly = ly - compv * 2 * liv
lysc = ly - sc * 16
IF lx > lg * 2 + 1 THEN GOTO torna1
IF ly > al * 2 + 1 THEN GOTO torna1
IF lysc > 16 OR lysc < 2 THEN GOTO salta2
LOCATE lysc, lx: PRINT ""
salta2: me$ = me$ + scrit$ (ly / 2, lx / 2)
torna1: NEXT
IF lysc > 16 THEN
sc = sc + 1: GOSUB tasto: GOSUB griglia: GOSUB compila
END IF: IF lx > lg * 2 + 1 THEN GOTO livelv1
IF ly > al * 2 + 1 THEN
LOCATE 18, 2: PRINT "messaggio tradotto "
GOSUB tasto1: GOTO scrive
END IF: GOTO livelo1
scrive:
CLS: PRINT UCASE$ (me$) : GOSUB tasto: RETURN

drive: ' ************** parte drive *****************
CLS: INPUT "Nome del file - messaggio "; nfi$
IF nfi$ = ""THEN RETURN
PRINT " Drive (0 / 1) ?": GOSUB tasto1
IF A$ = "1"THEN CHDIR "df1: ": ELSE CHDIR "df0: "
PRINT " INSERIRE IL DISCO ARCHIVIO "
PRINT " e premere un tasto": GOSUB tasto1
PRINT " ...attendere...": IF b = 5 THEN GOTO drive1
OPEN"i", #1, nfi$: INPUT#1, form$
IF form$ = "s"THEN
PRINT " Messaggio senza codice": GOSUB tasto: GOTO dri2
END IF: IF form$ < > "c"THEN
PRINT " Formato sconosciuto": GOSUB tasto1: RETURN
END IF: PRINT " Messaggio con codice": GOSUB tasto
INPUT#1, nc: FOR i = 0 TO nc: FOR i1 = 1 TO 2: INPUT#1, cod
(i, i1)
NEXT: NEXT: INPUT#1, compo: INPUT#1, compv
dri2: INPUT#1, al: INPUT#1, lg
FOR i = 1 TO al: FOR i1 = 1 TO lg: INPUT#1, scrit$ (i, i1)
NEXT: NEXT: CLOSE #1: ON ERROR GOTO 0
RETURN
drive1: ' ******
IF scrit$ (1, 1) = ""OR cod (0, 1) = 0 OR al = 0 THEN
PRINT "Nessun messaggio da salvare": END
END IF: PRINT " Save con codice ( c ) o senza ( s ) ?"
GOSUB tasto1: form$ = A$
IF form$ < > "c"AND form$ < > "s"THEN form$ = "c"
OPEN"o", #1, nfi$: PRINT#1, form$
IF form$ = "s"THEN PRINT " SAVE senza codice": GOTO dri3
PRINT " SAVE con codice"
PRINT#1, nc: FOR i = 0 TO nc: FOR i1 = 1 TO 2: PRINT#1, cod
(i, i1)
NEXT: NEXT: PRINT#1, compo: PRINT#1, compv
dri3: PRINT#1, al: PRINT#1, lg
FOR i = 1 TO al: FOR i1 = 1 TO lg: PRINT#1, scrit$ (i, i1)
NEXT: NEXT: CLOSE #1: ON ERROR GOTO 0
RETURN
' ********** subroutines varie *****************
griglia:
CLS: alris = al: IF al > 8 THEN al = 8
casx = xc * lg: casy = al * yc: LINE (xp, yp) - (casx + xp, casy +
yp), , b
FOR ig = 1 TO al: LINE (xp, yc * ig + yp) - (casx + xp, yc * ig +
yp)
NEXT: FOR ig = 1 TO lg:
LINE (xc * ig + xp, yp) - (xc * ig + xp, casy + yp) : NEXT
al = alris: RETURN
tasto:
LOCATE 19, 2: PRINT "TASTO"; SPACE$ (40)
tasto1: A$ = "": WHILE A$ = "": A$ = INKEY$: WEND: RETURN


code:
n = 0: IF al = 0 THEN
PRINT "Definisci prima la griglia": GOSUB tasto: RETURN
END IF: GOSUB griglia: LOCATE 18, 2: nc = 0: m = MOUSE (0)

PRINT "Segna col mouse il codice (max 20 car.) "
LINE (495, 156) - (545, 175), , b
LOCATE 19, 51: PRINT "FINE"
controllo:
IF MOUSE (0) < > 1 THEN FOR i = 1 TO 1500: NEXT: GOTO
controllo
cox = INT (MOUSE (1) / 10) + 1: coy = INT (MOUSE (2) / 9) + 1
IF cox > 50 AND cox < 54 AND coy = 19 THEN GOTO daticodice
IF (cox / 2) < > INT (cox / 2) OR cox > lg * 2 + 1 THEN GOTO
controllo
IF (coy / 2) < > INT (coy / 2) OR coy > al * 2 + 1 THEN GOTO
controllo
LOCATE coy, cox: PRINT "X": nc = nc + 1
cod (nc, 1) = cox: cod (nc, 2) = coy
IF cod (nc, 1) > cod (0, 1) THEN cod (0, 1) = cod (nc, 1)
IF cod (nc, 2) > cod (0, 2) THEN cod (0, 2) = cod (nc, 2)
IF nc = 20 THEN GOTO daticodice
GOTO controllo
daticodice:
LOCATE 18, 2: PRINT nc; "Caratteri"; SPACE$ (30)
GOSUB tasto: RETURN
livc: CLS ' ***********
INPUT "Livello di compenetrazione orizzontale (0) "; compo
INPUT "Livello di compenetrazione verticale (0) "; compv
RETURN
hl: ' *** chiede altezza e larghezza ******
CLS: INPUT "Altezza in caselle (max 32) "; al
INPUT "Larghezza in caselle (max 30) "; lg
IF al > 32 OR al < = 0 THEN GOTO hl
IF lg > 30 OR lg < = 0 THEN GOTO hl
RETURN
compila:
IF al < 8 THEN alris = al: ELSE alris = 8
FOR ig = 1 TO alris: FOR i1g = 1 TO lg: LOCATE ig * 2, i1g * 2
IF ig + 8 * sc < = 32 THEN PRINT scrit$ (ig + 8 * sc, i1g)
NEXT: NEXT: RETURN
calcolacar:
are = al * lg: lacox = cod (0, 1) / 2 - compo
lacoy = cod (0, 2) / 2 - compv: car = INT (are * nc / (lacox * lacoy)
)


RETURN
```

SETMAP

Le parole chiave di ogni comando sono rappresentate in maiuscolo e vanno (eventualmente) adoperate così come sono.
In minuscolo, sono invece riprodotti i parametri che vanno ridefiniti dall'utente.

Setmap

E' questo un altro dei comandi di AmigaDos non inclusi nella directory **C** del disco Workbench, ma non per questo di minore importanza.
Per l'esattezza, è memorizzato nella directory **System** del floppy fornito a corredo di Amiga, e può essere richiamato tanto da Workbench che da Shell (o Cli).

Il compito cui presiede è quello di consentire l'uso di tastiere nazionali diverse dallo standard americano.
Esigenza, questa, che si fa sentire nettamente anche in Italia, essendo in circolazione molte tastiere contenenti caratteri tipici della nostra lingua (vocali accentate), e con differente disposizione di alcuni simboli.

SETMAP tastiera

Il nome del comando, impartito da solo, provoca soltanto un sintetico **memo** sulla corretta sintassi da seguire. Tuttavia, in funzione della sua particolare localizzazione, qualche nota è d'obbligo.

Con questo parametro va precisato il tipo di tastiera da implementare.
Occorre far riferimento al nome convenzionale dato ad una serie di files memorizzati nella subdirectory **Keymaps** della directory **Devs** del disco di sistema.
In tale posizione sono rintracciabili i due files utili per noi... italioti: la mappa "**i**" imposterà la tastiera italiana, mentre "**usa1**" la tastiera anglosassone.
Chi volesse adoperare altri particolari tipi di caratteri, legati ad altrettanto estemporanee posizioni sulla tastiera, potrà trovare utili tutte le altre possibili mappature, presenti nella omonima subdirectory del disco Extras. C'è subito da precisare che, quando si impartisce (p. es.) un "**Setmap i**", il comando cercherà il file di nome "**i**" tassativamente nella directory prima descritta, e lo stesso dicasi qualora si ricorresse al più semplice doppio click da workbench.
In quest'ultimo caso si ricordi che, per selezionare il tipo di tastiera, è necessario clickare una sola volta sull'icona di Setmap, quindi scegliere **Info** dal menu Workbench ed inserire la stringa **Keymap=i** nel riquadro **Tool Types** dopo aver clickato su **Add**.
Per abilitare il tutto è necessario uscire dalla finestra Info mediante l'opzione **Save**.

Inutile aggiungere che, se la scelta non dovesse riguardare la mappa dei tasti italiana, ma quella Usa, l'istruzione da inserire nella Info dell'icona diverrebbe **Keymap=usa1**.

Adoperando il disco Workbench senza che ne sia stata alterata in alcun modo la configurazione, **Setmap** risulterà adoperabile da Shell con facilità. Basterà digitarne il nome seguito da quello della specifica mappa dei tasti.
Questo perchè, nella startup-sequence del disco, viene aggiunta la directory **System** al normale percorso di ricerca dei comandi, notoriamente incentrato sulla directory **C** oltre a quella corrente. Tuttavia, dopo aver aperto una finestra Shell, si provi a digitare al suo interno...
Type s:startup-sequence
...(supponendo sempre che si sia lanciato il sistema con il disco Workbench).

Se il computer è dotato di **tastiera italiana**, si noterà, verso la fine del batch file, come il comando Setmap sia espresso nella forma...

SYS:System/Setmap i

...ovvero con la precisazione del suo percorso completo. A parte l'ovvia considerazione che una simile "verbosità" è sempre consentita, vi è anche un motivo preciso: il comando **Path** (si veda **Amigafacile** n. 75), che inserisce la directory System nel percorso di ricerca, viene eseguito in una fase successiva del file.
Quindi, senza la completa precisazione del percorso, Setmap non verrebbe rintracciato dal sistema.

Tutto ciò porta ad una logica avvertenza: se si adopera un disco personalizzato, o si altera la startup - sequence del Workbench, o comunque si provvede a creare un batch file che imposti il tipo di tastiera, si presti molta attenzione al Path.

In alternativa, si può anche spostare il file Setmap **dalla** directory System **alla** C, rendendone però più difficoltoso il reperimento da Workbench, non essendo questa directory dotata di icona.

Utile soprattutto, quest'ultima procedura, se si utilizza un disco di lavoro espressamente dedicato all'ambiente Shell.

Quando adoperare Setmap

In linea di massima, il comando **Setmap** trova il suo impiego ideale nel file **startup-sequence**, in modo da impostare subito all'accensione (o al reset) di Amiga il tipo di tastiera desiderato. Da notare che, in mancanza di indicazione contraria, ovvero se non è presente alcun comando Setmap, per default viene assunta come valida la **tastiera americana**.
Per usi non strettamente personali, e quindi nell'impossibilità di stabilire a priori quale sarà la tastiera adottata dall'utente finale di un nostro disco, è consigliabile, piuttosto che adoperare "seccamente" Setmap, sottoporlo (sempre nell'ambito della startup-sequence) ad un input che ne condizioni la scelta, come per esempio adottato nei floppy **Amigazzetta**. Per ottenere qualcosa di simile, è sufficiente ricorrere ad una sequenza di istruzioni Dos così strutturata:

Ask "Tastiera italiana? (Y/N)"
If Warn
Setmap i
Else
Setmap Usa1
Endif

N.75
Assign
Copy
Date
Dir
Install
Path
Search
Sort

N.76
Caratt. Speciali
Delete
Format
Protect
Rename

N.77
Execute
Direttive Batch
If
Skip...lab
Quit

N.78
Cd
Ed
Break

N.79
Which
Newshell E Newcli
Ser: Par: E Prt:
Con: E Newcon:
Nil: E Raw:

Gia' pubblicati

A partire dal n.75 della rivista, si è cominciato ad esaminare in profondità il **Dos di Amiga**, rivolgendosi soprattutto (ma non solo) a chi si avvicina per la prima volta a questo computer.
Chiaro che, con il succedersi degli appuntamenti, l'integrazione tra quanto trattato in precedenza e gli argomenti affrontati in queste pagine si fa sempre più stretta, anche se viene volutamente mantenuta la maggiore autonomia possibile.
Per quei lettori che inizino solo ora a seguire la nostra rivista, è consigliabile (anche se non indispensabile) procurarsi tutti i numeri a partire dal già citato 75: si disporrà così di un completo compendio da affiancare alla consultazione del manuale, spesso fin troppo succinto nelle sue indicazioni.
In questa pagina, qundi, è riportato un elenco analitico di quanto già sviscerato nelle pagine della rubrica **Amigafacile.**.

ICONX

IconX

Questo comando, forse uno tra i più utili e in grado di facilitare al massimo anche chi predilige l'ambiente Workbench, di fatto è anche uno dei più ignorati, probabilmente a causa della poca chiarezza descrittiva del manuale. La sua azione è molto semplice: consente l'accesso ad un qualunque comando del Dos direttamente da Workbench, anche quando questo necessiti di una finestra Shell o Cli per essere eseguito. Quindi rientrano in questa categoria (per esempio) tutti i files contenuti nella directory **C**, normalmente irraggiungibili da Workbench perchè privi di icona, e perchè non potrebbero operare senza una finestra Dos attiva. Lo stesso dicasi per eventuali **batch files** che, allo stesso modo, potranno essere attivati con il doppio click tanto caro agli estimatori di **Intuition**.

Per ottenere questo risultato, è necessario però ricorrere a qualche manovra. IconX, infatti, non è un comando che svolge la sua azione con modalità simili a quelle cui si è abituiti, vale a dire impartendolo da Shell. Al contrario, va inserito proprio nel più tipico elemento del Workbench: l'icona. Più facile a farsi che a dirsi, chiariamone il funzionamento ricorrendo ad un esempio pratico, cogliendo l'occasione per adottare qualche "trucchetto" altrimenti icomprensibile alla luce di quanto esposto sul manuale.Per il nostro esperimento, proviamo a creare... qualcosa che in effetti c'è già: una implementazione del comando **Avail** da Workbench.

Il comando **IconX**, come certamente noto, fornisce indicazioni dettagliate sulla **memoria** di Amiga, segnalando quanta ne rimane libera, quanta è già impegnata ed il totale disponibile, il tutto considerando la suddivisione tra **ram Chip** e **ram Fast**.
Molto comodo da Shell o Cli, di norma viene sostituito in ambiente Workbench da altri programmi di pubblico dominio, a meno che non ci si accontenti della più generica indicazione fornita sulla barra dei menu.
Per vederlo in azione a suon di mouse, occorre anzitutto preparare un file batch che lo invochi, fornirlo di icona, e specificare che deve essere adoperato Iconx.
E allora: si lanci il sistema con il floppy Workbench, se ne apra l'icona disco, e si "biclicki" su Shell. Tutto è pronto per iniziare i lavori.
Cominciamo col procurare un'icona al comando, che, attenzione dovrà obbligatoriamente essere di tipo **Project**.
Senza andare troppo lontano, ne abbiamo già una a portata di mano, quella di Shell, come si può constatare clickandovi sopra una sola volta e scegliendo **Info** dal Menu Workbench.
Per semplificare le cose, collocheremo il tutto in **Ram Disk**, quindi, dopo aver attivato la finestra Shell (= premuto il pulsante sinistro del mouse mentre il puntatore è al suo interno), si digiti
Copy Shell.info Ram:
Giusto per controllare, si apra l'icona Ram Disk: al suo interno dovrà essere visibile una copia dell'icona Shell. Già che ci siamo, cominciamo con l'assegnarle il nome con il quale la useremo, **XAvail**, ricorrendo all'opzione **Rename** del menu Workbench.
Ora, prepariamo il file batch che ci serve, che dovrà ovviamente chiamarsi **Xavail** per collegarsi all'icona appena trasferita, e trovarsi anch'esso in Ram Disk.
In pratica, si impartisca nella Shell un comando...

Copy * To Ram:Xavail
...quindi si copino queste tre righe...

.Key ""
Avail
Ask "Premi Return"

...premendo il Return dopo ogni riga, e adoperando **Ctrl+** dopo l'ultima (tecnica descritta su AmigaFacile del numero scorso). Se lo si preferisce, si può utilizzare **Ed**, o un altro Ascii editor di proprio gusto.
Esamineremo meglio il batch tra breve, per ora limitiamoci a constatare che, se tutto è in regola (si provi anche a listare la Ram per controllare), all'icona Xavail ora corrisponde realmente un file (di testo).
A questo punto, manca solo un'ultima modifica: si prema di nuovo (una sola volta) sull'icona di Xavail, quindi si selezioni Info.
Nella finestra apparsa, si potrà constatare la presenza del messaggio **Sys:system/cli** nel riquadro **Default Tool**. Questo, lo si sarà intuito, perchè l'cona era **prima** riferita al file Shell. Si clicki all'interno del riquadro Default Tool, si elimini quanto in esso contenuto (con **Amiga Destro + X** oppure adoperando il **Backspace**), e si digiti, al suo posto, **C:ICONX**. Per il momento, si esca dalla Info clickando su **Save**.
Ora si attivi nel modo consueto l'icona Xavail (doppio click): apparirà una nuova finestra di nome IconX, con al suo interno le informazioni di Avail, mentre un cursore attenderà la pressione del Return (mentre la finestra è attiva!) per chiudere la finestra dopo una breve attesa.
E' "quasi" quello che volevamo: il comando Avail è disponibile anche da Workbench, e si può copiare l'icona direttamente col mouse in un luogo più permanente, come un floppy, trasportandola direttamente col mouse.
Abbiamo aggiunto, però, un "quasi". E questo perchè si può ancora migliorare il tutto dandogli una dignità anche estetica, ed evitando il ritardo nella chiusura della finestra. Si riapra la Info dell'icona Xavail, e si clicki sull'opzione **Add** del riquadro **Tool Types**. Qui, possiamo ora inserire due parametri riguardanti la forma ed il nome della finestra, ed il ritardo nella sua chiusura. Tradotto in pratica, questo significa scrivervi (in maiuscolo!)...
WINDOW=CON:0/0/400/60/MEMORIA

...poi, dopo aver di nuovo clickato su **Add**...
DELAY=1
Si esca dalla info ancora con Save, e finalmente si riprovi a lanciare Xavail: tutta un'altra cosa, vero?
Le dimensioni della finestra, qui ottimali per uno schermo 80 colonne, possono naturalmente essere variate in funzione dell'output del comando da eseguire o di una differente risoluzione video.
Come misura supplementare, per velocizzare il tutto, si può anche rendere **residenti** i comandi del batch Xavail; si provi per esempio ad impartire nella linea di comando Shell:
Resident c:avail
...seguito da un:
Resident c:ask
...quindi si riattivi Xavail: l'esecuzione ne avrà decisamente guadagnato in velocità.
Torniamo al batch file. Questo, in linea teorica, potrebbe anche essere costituito dal solo comando Avail.
Aggiungendo **Ask**, però, si ottiene la facoltà di chiudere facilmente la finestra Dos aperta da IconX, mediante una pressione del Return.
IconX, però, non prevede la possibilità di un Input, ed ecco spiegato il motivo dell'istruzione **.Key ""** in testa al file comandi: così facendo, si potrà immettere qualcosa da tastiera (si veda anche *Direttive Batch* su AmigaFacile n. 77).
Quanto visto, può essere applicato con minime modifiche a qualunque altro comando.
Rieditando (stavolta con Ed) il file di testo Xavail, si può per esempio sostituire **Info** ad **Avail**, quindi modificare le dimensioni della finestra nella specifica **Window** dell'icona in modo che diventino **0/0/550/130**.
E non è che l'inizio...

LIST

Al comando List può essere associato un gran numero di parametri, alcuni costituiti esclusivamente da una **parola chiave**, altri da una parola chiave seguita da **dati** aggiuntivi a cura dell'utente. I parametri possono essere adoperati singolarmente, ma anche associati in qualunque forma per ottenere output il più "mirati" possibile.

List
Uno dei più classici tra i comandi del **Dos**, tuttavia spesso poco sfruttato in tutte le sue potanzialità.
Assieme al "cugino" **Dir**, svolge il compito di mostrare il contenuto di un disco, di una directory, o anche semplicemente alcune o tutte le caratteristiche di un singolo file, ivi compresi eventuali commenti ad esso collegati.
Nonostante l'apparente semplicità di uso, le caratteristiche di List legate all'AmigaDos possono riservare più di una sorpresa.
Non è affatto superfluo, quindi, descriverne tutte le possibili sfaccettature, spesso trascurate anche da chi non è più alle prime armi.

PAT pattern
Praticamente inutile. La parola chiave **Pat** può essere sostituita dalla sola lettera **P**, mentre **pattern** indica la selezione di ricorrenze che si desidera, basata sull'uso dei caratteri speciali di Amigados (si veda Amigafacile n. 76). Concretamente, gli stessi risultati si possono ottenere anche SENZA adoperare il parametro: con...
List Pat Di#?
...per esempio, verranno listati tutti i files della directory corrente che iniziano con i caratteri **Di** (diskx, dircl, disk.info, eccetera), ma un semplice **List Di#?** produrrà lo stesso identico output. Da dimenticarne l'esistenza.

Il nome di un floppy o della corrispondente unità periferica (**Disco:**, **df0:**, **df1:**, eccetera), o il nome di una **directory** eventualmente completa di percorso, o anche il **nome** di un singolo file. Senza la presenza di altri parametri, verranno fornite tutte le indicazioni possibili, vale a dire i nomi di files e directory, la dimensione in byte (solo per i file), lo stato dei bit di protezione (si veda **Protect** su AmigaFacile del n. 76), data ed ora di creazione o dell'ultimo aggiornamento, ed infine i commenti, se presenti.
Il nome può anche comprendere uno qualunque dei caratteri speciali, o **wildcards**, in grado di selezionare solo alcune ricorrenze. L'esempio più tipico è costituito da **List #?.info**, che elencherà solo i files-icona presenti nel percorso attuale.

KEYS

Oltre ai soliti dati, aggiungendo questo parametro viene visualizzato, di ogni file o directory, la posizione del suo **header block** sul dischetto.

Informazione, questa, utile soprattutto ai più esperti per eventuali manipolazioni a suon di Disk Editor.
Da considerare come una semplice curiosità.

List
Impartito da solo, provoca il Listing (guarda caso) della directory corrente. Poco da dire su di una operazione così elementare, se non ricordare che è possibile dirottare su carta (sempre, anche quando sono presenti altri parametri) l'output aggiungendo, dopo List, la redirezione...
List >prt:
Come ovvio, sostituendo a **prt:** il **nome** di un file, sarà questo a contenere l'output in Ascii del comando.
Attenzione, in quest'ultimo caso, ad evitare indesiderate **riscritture** (cancellazioni).
Come tutti i comandi di AmigaDos, è possibile far seguire a List il punto interrogativo (?), che mostrerà il cosiddetto **template**, ovvero un riassunto di tutti i parametri adoperabili, mentre il sistema resterà in attesa di un input da tastiera, che diventerà il parametro da assegnare.
Nel caso di List, il template è decisamente esuberante, occupando quasi due righe di una finestra Shell su 80 colonne.

Nohead

Finalizzato più che altro a particolari output su carta, si limita ad eliminare le informazioni iniziali e finali che il comandi List fornisce di default.
Queste indicazioni sono costituite da una riga iniziale tipo:
Directory df1: on saturday
...ed una finale che riporta il numero totale di files e directory, nonchè i blocchi usati.

TO file o device

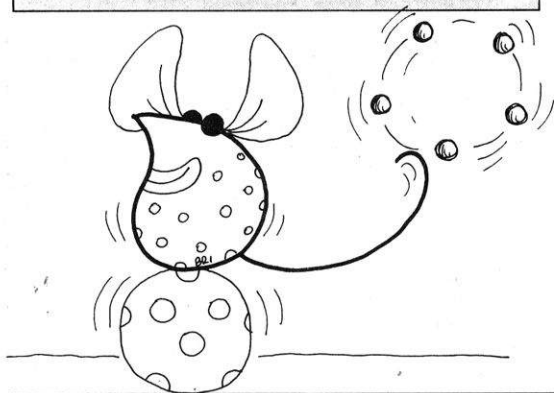
E' il parametro interno dedicato alla **redirezione** dell'output. In pratica si ottiene lo stesso effetto che utilizzando (per esempio) un già citato...
List >prt: Df1:
...che produrrà una stampa su carta del contenuto del floppy inserito nel secondo drive. La forma equivalente è:
List Df1: TO prt:.
Come ovvio, piuttosto che il device stampante si può utilizzare il nome di un file.
Nonostante la simiglianza, si presti attenzione ad una differenza sostanziale nell'uso delle due forme di redirezione. Il parametro **To**, seguito dal nome del file o del device, va utilizzato in coda alla sintassi di List, mentre la redirezione tramite il simbolo "**>**" va precisata **PRIMA** di qualunque altra specifica, subito a ridosso del comando List.

Quindi, per esempio, una forma...
List Df1: >file
...è scorretta (andrebbe impartito List >file Df1:), mentre con:
List Df1: To File
...tutto funzionerà per il meglio.

NODATES

Svolge esattamente il compito opposto di **Dates**, ovvero esclude dal listing data ed ora, rendendo più compatta la visualizzazione.

DATES

Con questa **keyword** (parola chiave), si induce **List** a visualizzare la data di creazione (o di aggiornamento) di files e directory. Di per sè non è molto utile, in quanto il comando provvede di default ad esibire questa informazione. In altre parole...
List Df1:
...e...
List Df1: Dates
...eseguiranno esattamente lo stesso compito. In certe occasioni, tuttavia, può anche esserne giustificata l'utilizzazione, soprattutto in associazione ad altri parametri. Se, per esempio, si adopera **Quick** (si veda descrizione), che lista i soli nomi senza altre specifiche, con...
List Quick Dates
...si potrà leggere la data (e l'ora)

QUICK

Come la traduzione di questa keyword lascia intuire, si tratta di un parametro che semplifica al massimo l'output, segnalando solo i nomi presenti nel device o directory cui il comando List è rivolto, senza neanche discriminare tra files e directory.

Per cui impartendo per esempio:
List sys: Quick
...con il disco Workbench adoperato per il boot, si avrà un output che accomuna senza distinzione Shell, S, Libs, Disk.info, eccetera.
Molto simile come risultato a quello che si potrebbe ottenere con il comando Dir, ma con il vantaggio che, associando a Quick uno degli altri possibili parametri di List, si può personalizzare notevolmente l'output associando ai nomi (p. es.) solo la data (Dates), o certe date (Since e Upto).

SUB stringa

Anche questo parametro riassume in sè le funzioni di una sequenza **wildcard**, nel senso che svolge lo stesso compito di alcuni caratteri speciali consentiti dal dos per definire precise ricorrenze. In particolare, aggiungendo Sub seguito da una stringa di caratteri, verranno listati tutti quei nomi che contengono al loro interno (in qualunque posizione) la suddetta stringa. Impartendo, per esempio...
List C: Sub le
...l'output mostrerà (se il sistema è stato bootato con il floppy Workbench) il listing dei file De**le**te e Fi**le**note (in entrambi è presente la sequenza **le** al loro interno). Lo stesso risultato, comunque, si potrebbe raggiungere con...
List C:#?le#?
Si badi al fatto che questo parametro, nella versione **1.2** del sistema, era implementato dalla sola lettera **S**, mentre dalla **1.3** è necessario adoperare Sub per esteso.

Files
Dirs

Molto comodi per una rapida ed efficiente selezione degli elementi da listare, questi due parametri limitano intuitivamente gli elementi visualizzati ai soli nomi di files (**Files**) oppure ai soli nomi di directory (**Dirs**).

Associabili a tutti gli altri parametri, sono addirittura adoperabili contemporaneamente. L'effetto finale, in questo caso, è quello di un normale List, ma in compenso... si sarà fatto un po' di esercizio fisico sulla tastiera (per masochisti congeniti).

LFORMAT=stringa

Tra tutti, è il più particolare dei parametri, in quanto consente al comando List di creare un file **script** (o **batch** che dir si voglia) adatto alle più svariate esigenze, evitando di dover digitare lunghe sequenze di comandi. In pratica, è necessario dirottare l'output di List verso un file, e quindi specificare nella stringa da porre dopo Lformat la funzione che intendiamo far svolgere ai comandi del nostro batch file.

Per esempio, supponiamo di voler creare un batch file che copi in Ram Disk tutti i programmi contenuti nella directory **Utilities** del disco WB (una copia del Workbench), purchè questi siano di data antecedente all'**1 gennaio 1990**.

A questo file batch, assegneremo il nome **Test**. Per crearlo, basterà impartire una (lunga...) istruzione...

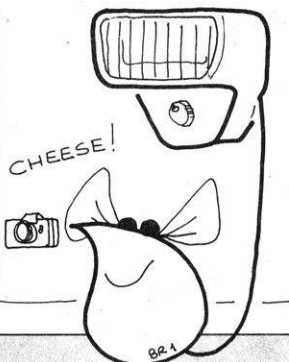
List WB:utilities Upto 01-Jan-90 To Ram:Test Lformat="Copy %S%S to Ram:"

Dopo l'esecuzione, si provi a digitare...

Type Ram:test

Si noterà la lunga sfilza di comandi **Copy** tutti correttamente rivolti alla **Ram**:. Questo file, è in pratica già pronto per essere mandato in esecuzione con **Execute Ram:test**. Si noti, nella lunga istruzione List, la presenza del simbolo **%S%S**, che verrà sostituito nel file batch dal nome del file listato, completo del suo percorso. Adoperando una sola ricorrenza **%S**, si sarebbe invece ottenuto il solo nome del file, senza l'indicazione del percorso.

SINCE data
UPTO data

Consentono di selezionare i nomi di files o directory in rapporto alla data. In particolare, adoperando **Since** verranno visualizzati tutti i files a partire dalla "data" specificata (compresa) in avanti, mentre con **Upto** la selezione includerà solo quei files o directory con data antecedente (o uguale) a quella inclusa nel parametro. Importante da sottolineare che la data va espressa nel formato Giorno - Mese - Anno, con **Giorno** composto da due cifre, **Mese** dalle prime tre lettere del corrispondente vocabolo inglese (Jan, Feb, Mar, eccetera), ed **Anno** espresso anch'esso con due cifre. Obbligatori i trattini che separano le varie componenti della data. In alternativa, il parametro data può anche essere espresso con **Today** (oggi), **Yesterday** (ieri) o **Tomorrow** (domani), o ancora specificando il giorno della settimana, naturalmente in inglese: Sunday per Domenica, Monday per Lunedì, e così via.

Un consiglio

Data la sovrabbondanza di parametri legati al comando List, può rivelarsi utile scegliere alcune tra le forme che si pensa di utilizzare maggiormente, ed assegnarle ad un "**alias**", in modo da non dover digitare lunghe e noiose sequenze da tastiera. Per esempio, il List accompagnato da **Nodates** è spesso utile per mantenere aperta una finestra dos di dimensioni ridotte, senza che l'output di List risulti disordinato e quasi illeggibile. In più, si potrebbero volere due comandi differenziati per leggere solo i nomi delle directory o solo i nomi dei files. In questo caso, tornerebbe utile impartire due istruzioni siffatte...

Alias Files List Files Nodates
Alias Dirs List Dirs Nodates

...ottenendo due comandi (Files e Dirs) direttamente impartibili da Shell, che provocheranno uno il list dei soli nomi dei files ed uno dei soli nomi di directoy, in entrambi i casi senza data.

Attenzione, però. I due "nuovi" comandi, seppur validi, potranno operare solo nella directory corrente, ovvero senza la possibilità di aggiungervi alcun parametro, come ad esempio l'indicazione di un device o di una directory. Quindi, in definitiva, si potrà usare Dirs (o Files) senza alcuna aggiunta, mentre **Dirs df1**: provocherà un errore di valutazione nel sistema, senza alcun risultato pratico (lo si provi, è sempre meglio toccare con mano).

di Lorenzo Emilitri

AMIGA IN MUSICA

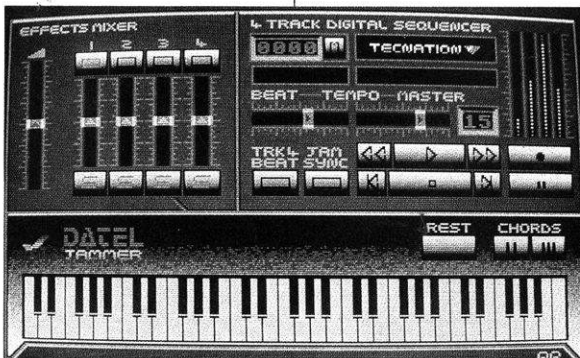
Continuiamo il viaggio che ci porterà al cuore della musica di Amiga, affrontando altri due programmi che ogni musicista elettronico deve conoscere

Oktalyzer è un programma piuttosto conosciuto tra gli appassionati, ed ha avuto subito una buona diffusione grazie alla capacità di suonare sino a **8 note contemporaneamente**. Ma l'etichetta di **Super Sound Tracker** che gli è stata appiccicata sin dalla sua comparsa sul mercato non rende giustizia alla potenzialità intrinseca di questo programma: potremmo dire che Oktalyzer rappresenta, da una parte, la logica evoluzione delle idee che stanno alla base del Sound Tracker, e, dall'altra, un ambiente di lavoro altamente professionale in grado di rivaleggiare con le potenzialità di analoghi programmi disponibili solo su computer più potenti (e, inevitabilmente, più costosi). Sin dal primo caricamento potremo ammirare, nella parte alta dello schermo, i numerosi comandi disponibili, mentre al centro, grazie all'uso di un font piuttosto piccolo ma leggibile, si scorgebuona parte del **pattern** corrente. Alla prima occhiata si nota che i comandi e gli indicatori sono raggruppati in quattro sezioni distinte: "Actual Song", "Editor", "Actual Sample" e "Memory".

Actual Song

In questa sezione si trovano sei comandi: oltre ai consueti **New** (cancella la canzone, ma non i samples), **Load** e **Save** (da notare che Oktalyzer salva sempre e solo **moduli**), vi sono ben due modi di uscita (**Exit**, uscita normale, e **CLI**, ovvero ritorno al WorkBench mantenendo attiva un'icona che permette di rientrare in Oktalyzer); dulcis in fundo la possibilità (tramite **Pref**) di personalizzare l'ambiente di lavoro. Oltre alle scelte più consuete (significato dei **tasti** del tastierino, **colori** e **autorepeat**), possiamo scegliere la configurazione di ciascun canale audio, la lunghezza dei pattern, il modo in cui devono essere caricati i samples e il modo in cui devono essere salvati. Notevole la possibilità di modificare tutti i parametri numerici semplicemente premendo i due tasti del mouse, rispettivamente per incrementare o decrementare valori selezionati. Alla destra delle opzioni si trovano i tre parametri **Pos**, **Patt** e **Len** già note a chi usa SoundTracker. Il loro uso è comunque facilitato da altre due opzioni, **Ins** e **Del**, che permettono di inserire (o cancellare) un certo pattern in un punto specifico della canzone, senza per questo modificare i valori di Patt successivi. Altri due indicatori ricordano la lunghezza della song in pattern (**Slen**) e la velocità con cui il pattern viene eseguito (**Speed**).

Editor

Nella sezione Editor si trovano comandi per l'esecuzione del brano (**Play Song** e **Play Pattern**), comandi di editing di singole note e comandi per l'editing di interi blocchi. Tra i primi troviamo **Edit** (è uno switch che permette, o meno, l'inserimento delle note; utile per non alterare inavvertitamente il nostro lavoro), **Poly**

Oktalyzer, istruzioni per l'uso

Ad ogni voce di Oktalyzer corrisponde una sequenza del genere...

C-1 1000

I primi due caratteri, a partire da sinistra, indicano la nota utilizzata, mentre il terzo è l'ottava della nota stessa. Quindi, ad esempio, **C-1** vuol dire **Do1**, e **F#2** vuol dire **Fa Diesis** della terza ottava. Il quarto carattere è il numero di strumento utilizzato; può essere un numero da **1** a **9** o una lettera maiuscola dalla **A** alla **Z**. Il quinto carattere è il comando opzionale che può essere inserito; lo **0** indica che non vogliamo utilizzarne alcuno. Per l'elenco dei comandi riconosciuti, vedi l'articolo. La sesta e la settima cifra rappresentano i parametri del comando; tranne che nell'arpeggio, dove l'una è indipendente dall'altra, formano un numero a 8 bit scritto in esadecimale.

Quindi, il comando **V40** vuole dire "imposta il volume a $40 (=64)", il comando **S05** vuol dire "lascia 5/50 di secondo tra una riga e l'altra del pattern che stai suonando" e così via.

I comandi di Oktalizer

(permette di inserire le note utilizzando più voci), **Quant** (permette di definire la lunghezza fissa delle note; ad esempio, settandolo al valore **4**, dopo aver inserito una nota il cursore si troverà quattro spazi sotto di essa, risparmiando l'uso del mouse e/o dei cursori), **Plen** (permette di definire la lunghezza del pattern corrente) e **Midiedit** (che permette l'inserimento delle note tramite uno **strumento Midi** opportunamente collegato).

Prima di descrivere i comandi per l'editing di blocchi di note, è necessario specificare come si definisce un blocco.

Per fare ciò è necessario portare il cursore ad uno degli angoli del blocco (che è ovviamente rettangolare), premere uno dei tasti **Shift** e spostarsi tramite i **tasti cursore** all'angolo opposto, sempre tenendo Shift premuto. Per eliminare il blocco è necessario di nuovo usare Shift e cursori, ma in maniera diametralmente opposta. I comandi relativi sono: **Copy** (copia il blocco selezionato in un buffer Ram) attivabile anche con il tasto **F4**; **Cut** (cancella tutto il blocco); **Replc** (copia dal buffer Ram al pattern nel punto indicato dal cursore) anche con il tasto **F5**, **Mixit** (aggiungi al pattern, nel punto indicato dal cursore, le note presenti in memoria), **Noteup** e **Notedown** (innalza o abbassa di un semitono tutto il blocco), **Oktaup** e **Okdown** (idem, ma per le ottave), **Chinst** (sostituisce uno strumento con un altro) e **Chaeff** (sostituisce un effetto con un altro).

Actual Sample

Ancora più a destra troviamo la sezione Actual Sample: sono disponibili ben 36 samples (da **0** a **9** e da **A** a **Z**) contemporaneamente in memoria, anche più lunghi di 32k, in tre modi diversi.

Si usa il valore **4** se il sample deve essere suonato da solo su un canale e **8** oppure **B** se il sample deve essere suonato assieme ad un altro su un canale. Infatti, le 8 voci sono ottenute suonando due samples contemporaneamente sullo stesso canale, grazie all'intervento fuori programma del 68000. La differenza tra il modo "8" e il modo "B" è che il primo non implementa i parametri **Rstr**, **Rlen** e **Vol** che ormai conosciamo benissimo (in caso contrario, è meglio rileggere l'articolo "SoundTracker"). Sempre nella sezione Actual Sample, oltre ad opzioni quali **Load**, **Save**, **Clear**, **Clear All**, **Swap** (scambia di posto in memoria due suoni), **Copy**, **Mix** e **Clrcb** (CLeaR Copy Buffer), troviamo un piccolo ma efficiente campionatore, con tanto di **filtri**, **dump** della forma d'onda, controllo del **pitch** e del **volume**, per non parlare di altre opzioni più "classiche" quali **Cut** e **Copy**. E' inoltre possibile ascoltare lo strumento selezionato in maniera polifonica.

Memory

Le ultime opzioni da esaminare appartengono alla sezione Memory (lunghezza della canzone corrente e riserva rimasta di memoria **Chip** e di **Fast**) e il controllore dei canali: l'asterisco (*) indica che il canale è attivo, mentre un simbolo grafico appare e scompare in fase di esecuzione ogniqualvolta viene suonata una nota. Passiamo ora al **Pattern**: questo viene selezionato premendo il tasto **Alt** più il cursore destro / sinistro. L'inserimento delle note avviene alla stessa maniera che nel SoundTracker (anche se molto più comodamente, grazie all'apprezzato intervento del "topo"). Le uniche differenze di rilievo sono nei comandi: essi non sono più un *numero* ma una *lettera*, ed hanno i seguenti significati...

A: Arpeggio;

D: Slide verso il basso del numero di semitoni indicato;

H: Portamento su;

L: Portamento giù;

S: Velocità (espressa in cinquantesimi di secondo);

U: Slide verso l'alto del numero di semitoni indicato;

V: Volume.

(Per maggiori informazioni su come inserire i comandi, riferirsi al riquadro specifico).

Il programma non presenta solo aspetti positivi: prima di tutto, è sin troppo facile scrivere canzoni bellissime ma di lunghezza a dir poco mostruosa (il tetto dei **200 k** viene sfondato con estre-

Note di "Tu Scendi dalle Stelle"

Per ogni riga il primo valore si riferise al numero di spazi verticali da lasciare tra una nota e l'altra. Subito dopo sono indicate le quattro note (accordo) da inserire in fila orizzontalmente, magari aiutandosi settando opportunamente **Poly**.

1 D-2 B-2 G-2 G-1
3 D-2 B-2 G-2 G-1
1 E-2 C-2 A-2 F#2
3 D-2 B-2 G-2 G-1
1 C-2 A-2 F#2 D-1
1 C-2 A-2 F#2 D-1
7 B-1 G-2 D-2 G-1

Tabella 1
Note da usare con Octalyzer

ma facilità e naturalezza); inoltre, la mancanza di una struttura software come la **PresetList** obbliga il programma a salvare sempre e solo moduli, e ciò significa che per scrivere una canzone completa è spesso necessario utilizzare più di un disco per tutti i passaggi intermedi. Per non parlare del fatto che, pur essendo teoricamente compatibile con i moduli del SoundTracker, nel 90% dei casi appare un requester del genere "**Not Possible!**" e il modulo viene rifiutato. Per finire, il replayer a 8 voci è praticamente introvabile e "mangia" una quantità oscena di tempo macchina, mentre quello a 4 voci esiste solo in versione DevPac.

Future composer

Future-Composer rappresenta il tentativo di superare il principale limite di Oktalizer (e di Soudtracker): l'elefantiaca occupazione di memoria, causata dall'uso dei samples. Ciò è ottenuto tramite l'uso di suoni "sintetici", ovvero di forme d'onda fisse che vengono modificate dinamicamente nel corso del tempo. Questo porta alla creazione di suoni molto particolari e di effetti di sicuro effetto (scusate il bisticcio), per non parlare della possibilità di ottenere un **decay** decente senza dover per questo allocare qualche decina di kilobyte. Future Composer è stato scritto da SuperSero dei Superions basandosi sul replayer della nota software house Thalion, che ha origine nel replayer scritto da David Whittaker per la musica del gioco Astaroth. E' sicuramente il programma più "difficile" tra quelli presentati, ma, se scrivete musiche per intros o demos, il gioco vale la candela... una musica relativamente lunga che utilizzi anche samples occupa non più di 30k, e non è così difficile scrivere una bella musica rimanendo entro i 10k di modulo. Future-Composer si presenta con tre schermi separati: lo schermo principale (che ricorda un po' il Rockmonitor e i vari Soundmonitor del C-64), il pattern editor e il sound editor.

Esamineremo per primo il sound editor: ogni suono è composto da una sequenza di modulazione del volume che a sua volta si rifà ad una sequenza di modulazione della frequenza che utilizza **Samples** (ve ne possono essere 10 in memoria, lunghi fino a 32k) o **forme** d'onda predefinite; sono disponibili 15 onde sinusoidali, 23 onde quadre e 2 onde a dente di sega. Non ci addentreremo nei meandri della programmazione degli strumenti, in quanto richiederebbe molto (troppo) tempo; a nostra parziale discolpa possiamo dire che Future Composer è (o almeno dovrebbe essere) sempre accompagnato da un file di documentazione molto chiaro ed altrettanto lungo. Esaminiamo ora il **Pattern Editor**: in Future Composer, per risparmiare memoria, i pattern sono lunghi **32** posizioni e larghi solo **una** voce. Se questo vi sembra al momento una cosa stupida, il mistero si svelerà quando affronteremo lo schermo principale.

L'editing avviene a grandi linee come nei programmi precedenti, solo è molto più difficile andare "ad orecchio" in assenza di uno schema molto preciso.

Per selezionare i pattern, così come gli strumenti, usate i tasti **7/8**, **4/5**, **1/2** e **0/.** del tastierino numerico. Inoltre, bisogna ammettere che l'editing è piuttosto scomodo: l'inserimento delle note avviene solo su una ottava (quella col **Do** corrispondente al tasto **Q**), e per selezionare le varie ottave è necessario premere i tasti funzione **F1..F4**.

La nota è indicata alla solita maniera, mentre è seguita solo da due cifre (e non quattro come nel ST o in Oktalyzer) che contengono il numero dello strumento usato, da **$00** a **$3F**, più informazioni sul portamento (vedi il file di documentazione).

Lo schermo principale è diviso in quattro colonne grandi, più una piccola; ognuna delle quattro è a sua volta divisa in sei cifre che rappresentano (nell'ordi-

F. - Composer: le note

Per ogni riga, prima è contenuto il numero di **spazi verticali** da lasciare tra una nota e l'altra, e poi le quattro note (**accordo**) da inserire in fila orizzontalmente. Le note qui riportate vanno inserite due volte.

2 G-2 E-2 C-2 C-1
0 A-2 F-2 C-2 C-1
1 G-2 E-2 C-2 C-1
5 E-2 C-2 G-1 C-1

Le note di "Bianco Natal"

Oktalyzer e "Tu scendi dalle stelle"

Per prima cosa, caricato il programma, dobbiamo cofigurarlo secondo le nostre esigenze, cioè a **4** canali. Per fare ciò, scegliamo **Pref** e premiamo, con il pulsante destro del mouse, sugli asterischi posti vicino al messaggio **Channel Config** fino a che ne rimangono solo quattro, degli otto presenti all'inizio. Per uscire dal menu premiamo **Use**.

Successivamente carichiamo un campione di **pianoforte**, che supponiamo si chiami semplicemente **Piano** (e che, ovviamente, deve esser presente su disco), segliendo l'opzione **Load** dai comandi dedicati alla gestione degli strumenti (colonna di **Actual Sample**); apparirà un normalissimo requester per file, e di lì dovremo scegliere il campione da caricare (non caricate files di testo o altro, perchè il sistema tenterà di "suonarli" e non hanno un suono molto gradevole...). Selezionato il sample prescelto (cliccando su **OK**), il sistema lo caricherà come **sample 1**, nome che verrà riportato nella colonna di Actual Sample, dopo il caricamento.

Prima di inserire le note, scegliamo la **velocità**, modificando, con i due pulsanti del mouse, il valore di **Speed** sul valore **A**; successivamente abilitiamo l'editing da tastiera (**Edit**, della colonna **Editor**, su **On**). Tramite l'uso del mouse, dei cursori e dei tasti (ad ognuno dei quali è associata una nota) inseriamo le note presentate nella **tabella 1**. Per quanto riguarda i tasti, essi sono disposti come su un pianoforte. Ricordiamo che il tasto **Q** è il **Do2** mentre **Z** è il **Do1**.

Per ascoltare la nostra creazione, è sufficiente utilizzare **Play Song** oppure **Play Pattern**.

Future-Composer e Bianco Natale

Per prima cosa è consigliabile caricare un **sample**, piuttosto che la creazione di uno strumento sintetico, perchè è una procedura più facile e veloce. Per fare ciò, clickiamo sull'icona **Type** fino a che non appaia la scritta **Sample** vicino all'indicazione **File-type**. Quindi carichiamo il sample vero e proprio, utilizzando il requester presente in basso a destra. Per ogni livello di directory è necessario premere la scritta **Dir** perchè i titoli siano caricati. Trovato il giusto sample, scegliamolo premendo **Load**.

Il fatto che il sample sia stato caricato non significa che sia riconosciuto dal programma; per fare ciò scegliamo **Edit Sounds** ed inseriamo la sequenza "**01 00 00 00 00 40 E1**" nella prima sequenza di modulazione del volume; premendo **7** sul tastierino numerico ed inseriamo un'altra sequenza ("**01 00 00 00 00 18 E1**") nella seconda sequenza di modulazione del volume. In seguito, si sceglie la sequenza di modulazione della frequenza (il riquadro posto sotto quello della modulazione di volume) ed inseriamo la sequenza "**E2 00 00 E1**".

Quindi scegliamo **Back to Main Screen**, e **Edit Patterns**.

Nel pattern editor, inseriamo le note contenute nella **tabella 2**; per selezionare l'ottava **1** / **2** premiamo, rispettivamente, **F1** ed **F2**, mentre per muoverci tra i pattern sono sufficienti i tasti cursore. Inserita la tabella 2, con **Return to Main Screen** inseriamo il valore **00** vicino a **Last Step**.

Successivamente inseriamo la stringa "**00 00 00, 01 00 01, 02 00 01, 03 00 01, 0C**" nel sequencer posto al centro dello schermo.

A questo punto, con **Play**, le magiche note di Bianco Natal si diffonderanno per l'aere!

Future - Composer in musica

ne, partendo da sinistra, in gruppi di due cifre): il numero di pattern, il valore di trasposizione musicale e il valore di trasposizione sonora.

E' necessario ora aprire una breve parentesi: non solo si è svelato il motivo per cui ogni pattern è largo solo una voce, ma anche quello che ha spinto gli autori a preferire tanti pattern brevi a pochi pattern lunghi.

Infatti il valore di trasposizione musicale (espresso in **semitoni**, in notazione esadecimale con segno) viene sommato a tutte le note del pattern, rendendo inutile l'inserimento di due pattern diversi se tutte le note in essi contenute differiscono di un egual numero di semitoni.

Lo stesso avviene per il valore di trasposizione sonora e gli strumenti, ed il risultato finale è il risparmio di preziosissima Ram. Naturalmente, sta al compositore utilizzare al meglio le opzioni.

Nella colonna più piccola che si trova a destra, è specificata la velocità a cui devono essere eseguiti i pattern. Essa è per default **3/50** di secondo, ma può essere variata a piacere. Nella parte inferiore dello schermo vediamo altri parametri: **First Step** è la prima riga dello schermo principale che sarà eseguita, **Last Step** è l'ultima e **Current Step** quella attuale. Sotto si trovano i gadget per i 10 samples caricabili dall'utente; chiaramente, è un obbligo morale, utilizzando questo programma, limitare i samples a casi di assoluta necessità ed utilizzare in tutti gli altri casi strumenti "sintetici". Attenzione però: una musica nel cui modulo non sia presente nemmeno un sample verrà eseguita dal replayer, ma non caricata dall'editor.

Se, quindi, non avete intenzione di usare nessun sample, vi conviene caricare come sample 0 un file di testo lungo poche decine di bytes con il vostro nome e il titolo della musica e poi ignorarlo mentre progettate gli strumenti; solo così il vostro modulo potrà essere (ri)caricato e modificato una volta immagazzinato.

Inoltre, il replayer v1.00 ignora modifiche del valore nella colonna della velocità, per cui usate una sola velocità (o scrivete un replayer più funzionale!).

Future-Composer è un programma che può dare grandi soddisfazioni, ma richiede un impegno, un livello di conoscenza della musica, della fisica del suono ed una dedizione fuori dal comune. Se pensate di avere la testa abbastanza dura, provatelo e non fatevi scoraggiare dai primi inevitabili insuccessi.

Breve glossario.

Arpeggio: serve a simulare gli accordi, incrementando la nota da suonare di tanti semitoni quanto vale la prima cifra dei parametri; lo stesso viene fatto, in seguito, con la seconda cifra prima che la nota corrente vengae ripristinata.

Decay: è il periodo di tempo in cui una nota suonata da uno strumento, dopo aver raggiunto il volume massimo, raggiunge il volume 0. La maggior parte dei suoni campionati mancano di questa parte, cioè il suono è campionato per circa un secondo o due, solo nella fase iniziale.

Midi: è lo standard di comunicazione che permette il passaggio di dati e suoni tra computer e strumenti musicali; oggi quasi tutti gli strumenti musicali elettronici sono Midi-compatibili;

Pattern: è una sequenza di note fissa che forma la base della musica su computer; di fatto, il computer vede una musica come una sequenza di pattern da suonare nell'ordine impostato.

Portamento: questa opzione rende possibile la modifica dell'altezza di una nota *mentre* viene suonata; in pratica, ciò significa poter "tirar su" o "tirar giù" una nota di tanti semitoni quanto vale il numero che segue il comando stesso.

Sample: è un suono trasformato in una serie di numeri e pronto per essere utilizzato da Amiga.

Slide: è lo "scivolio" continuo di una nota, o del volume, verso l'alto o verso il basso.

di Dario Pistella

IL GIOCO DEL DOMINO PER AMIGA E C/64

L'implementazione di giochi classici (e soprattutto semplici) rappresentano una "palestra" validissima per i programmatori principianti

Il breve programma presentato in queste pagine (valido per **qualsiasi** computer Commodore, ad eccezione dei PC compatibili), rappresenta la trasposizione (in gergo: **implementazione**) su computer del celeberrimo gioco del domino.

Domino, come si gioca

Per chi non lo sapesse, il Domino utilizza un certo numero di **pedine**, ognuna delle quali è formata da due quadratini che rappresentano, ciascuno, due valori numerici (compresi tra 1 e 9). I due (o più) giocatori, dopo aver "mescolato" le pedine (disposte sul dorso per impedire, ovviamente, di esaminarle) se le dividono in parti eguali decidendo se lasciarne alcune di riserva (per l'eventuale, successiva "pesca") oppure se non lasciarne alcuna.

Il gioco consiste nel disporre, alternativamente, le proprie pedine sistemandole al fianco di altre, già "scoperte", in modo che due numeri uguali si trovino adiacenti. Per esempio, una possibile sequenza è: **23**, **35**, **57**, ... continuando a turno ad affiancare pedine a destra o a sinistra indifferentemente.

Ad un certo punto del gioco, uno dei due giocatori non avrà più la possibilità di affiancare le proprie pedine e sarà quindi sconfitto; se, invece, è disponibile la "riserva", prenderà una pedina alla volta, finchè non ne preleverà una valida per la disposizione sul tavolo. Vincerà, anche in questo caso, chi per primo esaurirà le proprie pedine.

Giocando

All'inizio del gioco il computer mostra, nelle prime due righe dello schermo, le pedine possedute dal giocatore, sistemate in verticale. In basso verranno, invece, visualizzate due linee bianche che rappresentano le pedine del computer (rovesciate). A mano a mano che le utilizzerà, l'elaboratore provvederà a cancellarle dallo schermo.

Per comunicare al computer la propria scelta, bisognerà inserire, come **primo** numero della pedina, quello che si intende affiancare alle pedine già presenti sul video, seguito dalla **seconda** cifra che compone la pedina stessa e la **direzione** lungo la quale si desidera porla (**D** per destra oppure **S** per sinistra).

Provvederà il computer a spostare la pedina al posto giusto tramite istruzioni **Poke** (caso del C/64) seguite dalla locazione video a seconda del tipo di computer che si sta usando. Tale riconoscimento viene eseguito dalla prima riga del programma, poichè dopo l'esecuzione della breve routine presente sulle **ROM del C/64/128/16/+4** posta a partire dalla locazione 65517, la locazione di memoria **781** conterrà il valore 40 solo nel caso in cui il computer usato sia il C/64 o C/128.

Le pedine sono immagazzinate nella variabile **V(a,b)** dove **a** e **b** rappresentano i numeri che la compongono.

Nel caso usiate un **Amiga**, il listato di queste pagine è fin troppo chiaro per aggiungere altri commenti. Anzi, i possessori del miserello C/64 saranno costretti ad ammettere la notevole differenza (in meglio, si intende...) esistente tra la comprensibilità di AmigaBasic e quelle offerta dal più *matusa* Basic del C/64.

Miglioratelo, ragazzi!

Il programma, come ogni listato che si rispetti, è suscettibile di variazioni ed ampliamenti di vario genere, tra cui:

* una miglior **resa** grafica (densa, magari, di sprite o altre piacevolezze)
* miglioramento della routine che gestisce la risposta dell'elaboratore.
* stesura di una procedura che si preoccupi di scrivere il valore percentuale rappresentante la frequenza di uscita di ciascun numero.
* utilizzo del mouse per indicare, e spostare, la pedina.
* annullamento dell'ultima mossa.
* richiesta di suggerimento.
* e così via, all'infinito.

A voi, quindi, il compito di migliorare il programma usando tutta la vostra abilità di programmatori.

```
REM Domino (Versione Amiga)
REM by Dario Pistella
RANDOMIZE TIMER
inizializzazione:
FOR a = 0 TO 9:FOR b = a TO 9
c=INT(2*RND):c=c+1
assegnazione:
v(a,b) = c:d(c) = d(c) + 1
IF d(c) > 28 THEN c = 3 - c:GOTO assegnazione
NEXT b,a:g = 1:IF d(2) > d(1) THEN g = 2
eliminazione:
d = INT(9 * RND):e = INT(9 * RND)
IF (v(d,e) <> g) OR (d = 9 AND e = 9) THEN eliminazione
v(d,e) = 0:CLS:PRINT "eliminato:"d;e
REM *** stampa pezzi giocatore ***
FOR a = 0 TO 9:FOR b = 0 TO 9
IF v(a,b) = 1 THEN x = x+2:LOCATE 3,x:PRINT a:LOCA-
TE 4,x:PRINT b
y(a,b) = x
NEXT b,a
REM *** inizio ***
ln(1) = 15:ln(2) = 15
d(1) = 39:d(2) = 41:i(1) = -2:i(2) = 2:s(1) = 9:s(2) = 9:a =
9:b = 9:o = 1
IF v(9,9) = 2 THEN vars
GOTO cancella
giocatore:
LOCATE 1,50:PRINT"?   ":LOCATE 1,51
getakey:
a$ = INKEY$: IF a$ = "" THEN getakey
IF a$ = "a" THEN CLS:PRINT"ho vinto.":END
a = VAL(a$):PRINT a$;
get2:
b$ = INKEY$: IF b$="" THEN get2
b = VAL(b$):PRINT b$;:IF v(a,b) <> 1 AND v(b,a) <> 1
THEN giocatore
get3:
d$ = INKEY$
IF d$ <> "d" AND d$ <> "s" THEN get3
PRINT d$:o = 2:IF d$ = "s" THEN o = 1
IF s(o) <> a THEN giocatore

cancella:
z = y(a,b):IF z = 0 THEN z = y(b,a)
LOCATE 3,z:PRINT "  ":LOCATE 4,z:PRINT "  "
rigastampa:
IF d(o) > 73 THEN d(o)= 1:ln(o) = ln(o) + 1:GOTO stampa
IF d(o) < 2 THEN d(o) = 73:ln(o) = ln(o) - 1
stampa:
LOCATE ln(o),d(o):PRINT a:d(o)=d(o) + i(o):LOCATE
ln(o),d(o):PRINT b
v(a,b) = 0:v(b,a) = 0:d(o) = d(o) + i(o):s(o) = b
IF j THEN j = 0:GOTO giocatore

computer:
o = INT(2 * RND):o = o + 1
ciclo:
FOR a = 0 TO 9:FOR b = a TO 9
IF (a<>s(o) AND b<>s(o)) OR (v(a,b)<2 AND v(b,a)<2)
THEN NEXT b,a:GOTO fine
IF a <> s(o) THEN g = b:b = a:a = g
vars:
j = 1:GOTO rigastampa

fine:
IF t THEN CLS: PRINT"hai vinto":END
t = 1:o = 3 - o:GOTO ciclo
CLS:PRINT"ho vinto"
```

```
10 rem * ----------------- *
20 rem *       domino      *
30 rem * by dario pistella *
40 rem * ----------------- *
50 rem * per c/16-64-128-+4*
60 :
70 rem : determinazione computer :
80 :
90 sys65517:ifpeek(781)<>40thenl=2048
100 s$=chr$(157):c$=chr$(147):l=l+1024:p=l+799:d(1)=l+379:d(2)=l+380
110 rem : assegna pezzi :
120 :
130 fora=0to9:forb=ato9:c=int(2*rnd(0))+1
140 v(a,b)=c:c(c)=c(c)+1
150 ifc(c)>28thenc=3-c:goto140
160 nextb,a:g=1:ifc(2)>c(1)theng=2
170 rem : elimina il 55mo pezzo :
```

```
180 :
190 d=int(9*rnd(0)):e=int(9*rnd(0)):if(v(d,e)<>g)or(d=9ande=9)then190
200 v(d,e)=0:printc$"eliminato:"d;e:fork=1to1000:next
210 rem : stampa numeri :
220 :
230 printc$:lo=1:fora=0to9:forb=ato9
240 ifv(a,b)=1thenpokelo,48+a:pokelo+40,48+b:lo=lo+1
250 nextb,a
260 fork=p+1top+27:pokek,160:pokek+40,160:next
270 i(1)=-1:i(2)=1:s(1)=9:s(2)=9:a=9:b=9:o=1:ifv(9,9)=2then560
280 goto430
290 rem : giocatore :
300 :
310 printchr$(19)tab(30)"?   "s$s$s$;
320 geta$:ifa$=""then320
330 a=val(a$):printa$;:ifa$="a"then610
340 getb$:ifb$=""then340
350 ifb$="a"then610
360 b=val(b$):printb$;:ifv(a,b)<>1andv(b,a)<>1then310
370 getd$:ifd$<>"s"andd$<>"d"then370
380 printd$:o=2:ifd$="s"theno=1
390 ifd$="a"then610
400 ifs(o)<>athen310
410 rem : trova locaz. pezzo da canc. :
420 :
430 fork=1tol+27:ifpeek(k)-48=aandpeek(k+40)-48=bthen480
440 ifpeek(k)-48=bandpeek(k+40)-48=athen480
450 next:goto310
460 rem : stampa pezzo :
470 :
480 pokek,32:pokek+40,32:poked(o),a+48:d(o)=d(o)+i(o):poked(o),b+48
490 v(a,b)=0:v(b,a)=0:d(o)=d(o)+i(o):s(o)=b:ifjthenj=0:goto310
500 rem : computer :
510 :
520 o=int(2*rnd(0))+1
530 fora=0to9:forb=ato9:ifa<>s(o)andb<>s(o)thennextb,a:goto590
540 ifv(a,b)<2andv(b,a)<2thennextb,a:goto590
550 ifa<>s(o)theng=b:b=a:a=g
560 j=1:p=p+1:k=p:goto480
570 rem : messaggi :
580 :
590 iftthenprintc$"hai vinto":end
600 t=1:o=3-o:goto530
610 printc$"ho vinto"
620 end

ready.
```

Tra breve, su queste pagine, parleremo *sistematicamente* di...

MONDODOS:

Linguaggio C

Linguaggio Turbo Pascal

Linguaggio Quick Basic

Linguaggio Assembly 8088 / 80486

Sistema Operativo Ms - Dos

Hardware: schede e accessori

PIANETA AMIGA:

Linguaggio C (versione Amiga)

I Basic (interpretati e compilati)

Linguaggio Assembly 68000

Sistema Operativo Amiga Dos

Hardware: schede e accessori

La nostra testata vanta, da quasi un decennio, una notevole esperienza nel campo della didattica; moltissimi sono gli utenti del personal computer che devono le loro conoscenze all'impostazione della rivista che hai tra le mani.
Per venire incontro alle esigenze dei nuovi utenti, senza trascurare quelle dei suoi affezionati lettori (che, nel frattempo, si sono sensibilmente "evoluti") è stata presa la decisione di approfondire argomenti adeguati alle macchine degli anni '90.

Una sfida al mese

UN MASTER MIND DA REINVENTARE

Siete riusciti a portare a termine, vincendola, la "sfida" lanciata sul numero 79? No? Poco male; stavolta, infatti, vi proponiamo...

Il popolare gioco **Master Mind** fu uno dei primi ad essere implementato su un personal computer, tanto che rappresentò un vero e proprio "esercizio" di programmazione per i neofiti. Sappiamo benissimo che più di una volta, sulle nostre stesse pagine, ci siamo rifiutati di pubblicare articoli e programmi che trattavano questo (obsoleto) argomento dal momento che, gira e rigira, i listati proposti erano tutti eguali tra loro.

Questo mese, tuttavia, vi proponiamo una "sfida" legata a Master Mind.

Si tratta, in parole povere, di scrivere un programma (in un **qualsiasi** linguaggio, ma solo per **Amiga** o elaboratori **Ms-Dos** compatibili) in grado di trattare in modo esauriente e didattico il gioco. Il listato, insomma, deve non solo esser completo, ma offrire un'opportunità didattica per divulgare tecniche di programmazione inconsuete o particolarmente efficaci. Per quanto riguarda la **completezza**, devono esser previsti, ad esempio, vari gradi di difficoltà; la possibilità di memorizzare (su disco) le sequenza delle combinazioni digitate dai vari utenti che si succedono alla tastiera (per realizzare, magari, una specie di test di intelligenza), la possibilità di far apparire, a scelta, quadratini colorati o "pezzi" grafici creati, a parte, con un pacchetto grafico; input controllati di particolare efficacia; contatore del tempo impiegato alla soluzione del gioco allo scopo di elaborare statistiche, eccetera. Insomma, per farla breve, bisogna inserire tutti gli accorgimenti che possano rappresentare lo Status of Art per ciò che riguarda Master Mind implementato nei calcolatori del 1991. Il listato, ovviamente, risulterà lunghissimo, tanto che **non** potremo pubblicarlo su rivista ma lo renderemo comunque disponibile sulla nostra **BBS**. I lettori che accetteranno la sfida, tuttavia, dovranno scrivere il programma in modo altamente **strutturato** affinchè sia possibile estrarre almeno **tre** subroutines, relative ad altrettanti segmenti di programma. Queste dovranno avere caratteristiche tali da essere impiegate in **qualsiasi** altro programma, senza apportare alcuna modifica. Ne consegue che tali subroutines dovranno avere le caratteristiche di **universalità**, **chiarezza** e, soprattutto, **brevità** (ciascuna non dovrà occupare più di una pagina di rivista, comprese le Rem).

Il lavoro migliore verrà pubblicato e compensato con 150 mila lire. Agli autori di altri lavori, giudicati interessanti, verrà sottoscritto un abbonamento gratuito alla nostra pubblicazione.

In ogni caso, non ci stancheremo di ricordarlo, **prima** di inviare il dischetto, gli autori dovranno **telefonare** in Redazione (**solo** il giovedì pomeriggio) per stabilire se il programma risponde effettivamente alle caratteristiche richieste.

POSTAMIGA

(a cura di Domenico Pavone)

Separati in casa

Quale configurazione PC-compatibile si può raggiungere con un Amiga 2000? Le schede in commercio valgono la spesa?
(Graziano '68 - Roma)
(Vito Catania - Catania)

Il problema della implementazione dell'**Ms-Dos** su Amiga, può assumere toni contrastanti.

Da un punto di vista fisico, c'è stata indubbiamente una notevole evoluzione proprio in questi mesi.

Oltre alla vecchia **Janus XT**, ed alla "cugina" superiore **AT** basata sul microprocessore 80286, la stessa Commodore sta per mettere in produzione (e probabilmente lo avrà già fatto quando leggerete queste righe) un versione "turbo" della XT, pare di velocità più che raddoppiata.

Per non parlare della Pc-board per Amiga 500, dalle prestazioni tutt'altro che disprezzabili, della quale si può leggere a parte.

Come se non bastasse, è già in circolazione una scheda (non Commodore) che monta un **80386** con clock a 16 Mhz, e sicuramente altra "robétta" verrà fuori entro breve tempo.

La possibilità di sfruttare anche il mondo Pc, insomma, esiste, e con sufficiente grado di affidabilità.

Tutta da verificare, invece, l'opportunità di un acquisto che, da un punto di vista economico, può rivelarsi di un certo impegno.

Il problema riguarda soprattutto l'uso cui si intende destinare una conformazione hardware comprendente una scheda Ms-dos.

Se si pensa a sviluppi professionali legati ad un Pc, oppure dedicati ad una programmazione approfondita in tale ambiente, allora (**ma è solo un nostro parere**) non c'è che una strada: acquistare un Personal, magari da affiancare al più che valido Amiga.

Se, invece, si considera l'eventualità di un più generico approccio all'Ms-dos, che vada al di là della mera emulazione software, per adoperarne qualche applicativo o a fini puramente conoscitivi, l'acquisto di una scheda potrebbe anche trovare una giustificazione.

Con un occhio, però, all'aspetto economico.

I "compatibili", infatti, costano sempre meno, e non avrebbe senso, per esempio, spendere una certa cifra per l'acquisto di una scheda AT, quando con sole duecento, trecento o anche quattrocento mila lire in più si può avere un computer, di quella categoria, nella sua configurazione completa.

Il miglior multitasking che si conosca resta sempre quello ottenuto con **DUE** computer (eh eh!).

Informarsi sui prezzi delle due possibili scelte, insomma, è d'obbligo.

Se, poi, anche il prezzo delle schede dovesse abbassarsi fino a diventare realmente competitivo, beh, il dubbio potrebbe diventare amletico...

Voglia di pointer

Vorrei fare in modo che in un mio programma il pointer del mouse abbia una certa forma, diversa dalla solita freccetta. Però di linguaggio macchina non ne capisco niente; è possibile farlo in basic?
(Guido Mancini - Verona)

Certamente. L'argomento, comunque, può essere affrontato in due modi. Se tutto ciò che si desidera è un pointer diverso dalla freccetta, ma sempre uguale in tutte le applicazioni, allora non è neanche necessario scomodare il basic: basta adoperare il programma **Preferences** presente nel disco Workbench, editare un pointer personalizzato e salvare il tutto in uscita dal programma.

Naturalmente, andrebbe poi utilizzato sempre quel disco per il boot del sistema.

Con un pizzico di programmazione, anche in basic, si può però ottenere molto di più, come ad esempio uno o più pointer (diversi tra loro) collegati all'attività di una specifica finestra. La tecnica per ottenere un simile risultato non è particolarmente difficile; tuttavia, chi è proprio alle prime armi, potrebbe incappare in qualche problema, legato soprattutto all'uso delle

librerie di sistema. Niente di insormontabile, comunque: si segua con attenzione ogni passaggio, e tutto risulterà più chiaro.

Giusto per conoscenza, occorre anzitutto dire che il pointer altro non è che uno sprite, lo **Sprite Numero 0**. La cosa è importante, ma vedremo tra breve come si può manipolare questa struttura anche senza conoscerne approfonditamente i dettagli. Quello che al momento ci interessa sapere, è che l'aspetto di questo particolare sprite può essere modificato ricorrendo ad una apposita funzione della libreria Intuition, di nome **SetPointer**. A questa funzione vanno associati 6 parametri, e più precisamente:

1) L'**indirizzo** della struttura Window alla quale si intende associare il pointer. Visto che ci stiamo riferendo al basic, questo indirizzo si traduce molto semplicemente nella funzione **Window(7)**, da assegnare ad una variabile a 32 bit (p. es. **Finestra& = Window(7)**) subito dopo aver aperto la finestra con il ben noto statement Window (si veda il manuale).

2) Un **secondo** indirizzo, che precisa a partire da quale locazione di memoria è memorizzata una struttura di dati che definisce la **forma** ed i **colori** dello sprite. Vedremo meglio tra breve, senza troppa teoria, le implicazioni pratiche necessarie per esprimere il parametro.

3) L'**altezza** dello sprite, espressa in pixel.

4) La sua **larghezza**, che non può, ovviamente, superare i **16** pixel, limite massimo consentito per qualunque sprite.

5 - 6) Due **offset** (**X** per la coordinata **orizzontale** ed **Y** per la **verticale**) che specifichino a quale punto dell'immagine corrisponde il pointer vero e proprio, ovvero il singolo pixel attivato dal pulsante sinistro del mouse. In particolare, con X ed Y uguali a **zero**, il pixel "attivatore" corrisponderà all'angolo superiore sinistro del pointer. Con X = **-7** lo stesso sarà posizionato al centro dello sprite, e con X = **-15** sulla estremità destra.

Per capire meglio come trattare questa serie di informazioni, mettiamole subito in pratica riferendoci al listato 1 di queste pagine, redatto in AmigaBasic. Lo si copi dunque nella finestra di editing dell'interprete (o con un qualunque altro in grado di trattare files in formato ascii), e lo si salvi debitamente. Prima di mandarlo in esecuzione, è necessario prestare attenzione al solito problema dei files "**.bmap**" cui il programma deve accedere. In questo caso, poichè dovranno essere adoperate le librerie **Exec** e **Intuition**, sarà necessario rendere "visibili" (=agibili) due files: **Exec.bmap** e **Intuition.bmap**. Cosa che ci obbliga ad una chiacchierata iniziale. Exec.bmap è già presente nella directory BasicDemos del disco Extras 1.3, mentre Intuition.bmap **no** (sadismo puro della Commodore?); per cui occorre crearlo.

Per facilitare il tutto, si proceda, a titolo di esempio, come segue, tenendo presente che al posto della **Ram:** può essere specificato un qualunque altro disco di lavoro. Anzitutto, (dopo aver resettato il computer) da Workbench si apra l'icona-disco **Extras 1.3**, ed al suo interno quella della directory **BasicDemos**. Con il mouse, si trasferisca l'icona del file **Exec.bmap** in Ram Disk sovrapponendola alla corrispondente icona-disco. Ora si attivi il programma **ConvertFD**, presente anch'esso nella directory Basicdemos, biclickandone l'icona; oppure, se l'interprete del basic è già in memoria, adoperando Open dal suo menu e poi digitando per esteso all'interno del requester:

Extras 1.3: basicdemos / convertfd

Dopo lo start, il programma chiederà in input il **nome** del file da convertire nell'equivalente con suffisso **.bmap**. Si risponda...

Extras 1.3: Fd1.3/Intuition_lib.fd

Alla successiva richiesta, riguardante il nome completo da assegnare al file, digitare...

Ram: Intuition.bmap

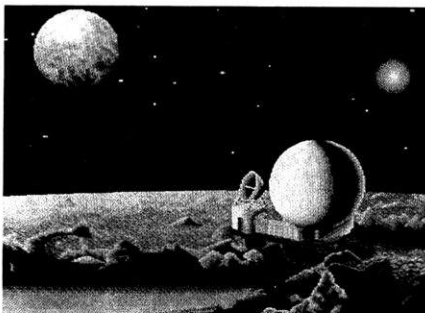
...con **Ram:** eventualmente sostituibile col nome del disco nel quale si vuole far finire il file, o anche l'unità periferica (**df1:**) se si dispone di un secondo drive.

Dopo una certa attesa (e dopo un intenso scambio di dischetti, nel caso si disponga di un solo drive e si insiste nel generare il file su un altro disco, anzichè su Ram:), al ricomparire dell'Ok nella finestra di output del basic, si potrà verificare, anche da workbench, la presenza del file, aprendo nuovamente l'icona Ram Disk (dopo averla richiusa, se già era aperta). Espletate queste noiose formalità, si può finalmente ricaricare il listato prima copiato, impartire in modo diretto...

Chdir "ram: "

...e finalmente lanciare il programma **demo** per il nuovo pointer. Se tutto è in regola, apparirà una finestra di dimensioni inferiori a quella di output, mentre il solito pointer sarà sostituito da un variopinto e stilizzato "**CCC**". Si provi, ora, a clickare all'esterno della finestra. Riecco la ben nota freccetta. Come controprova, si attivi di nuovo la finestra Demo.

Tirando le somme, il pointer personalizzato farà bella mostra di sè solo in realazione a quest'ultima. Nell'esempio citato, all'interno della finestra non viene svolta alcuna attività, se non quella di consentire l'uscita dal programma premendo un tasto; come intuibile, però, vi si può inserire qualunque propria creazione, addirittura implementando più pointer ognuno "riservato" a specifiche applicazioni.

Non ci resta che vedere più da vicino come funziona la routine, promettendo sin d'ora un omaggio ai meno pazienti: una **utility per rubacchiare il pointer da un dischetto**, ma soprattutto in grado di facilitarne la creazio-

```
'ASSEGNAZIONE DI UN POINTER DIVERSO
'AD UNA SPECIFICA FINESTRA VIDEO

'APERTURA LIBRERIE EXEC E INTUITION
LIBRARY "intuition.library"
LIBRARY "exec.library"
DECLARE FUNCTION Allocmem& LIBRARY

'ALLOCA BUFFER DI MEMORIA PER SPRITE
byte& = 100: chip& = 2
buf& = Allocmem&(byte&, chip&)
FOR x = 0 TO 71
READ a: POKE buf&+x, a
NEXT
'
WINDOW 2, "demo", (10, 10) - (400, 150), 18
PRINT "Per uscire dal programma
PRINT "premi un tasto quando"
PRINT "questa finestra è attiva"

' ATTIVA NUOVO POINTER
fin& = WINDOW(7): 'base struttura window
a& = 16: l& = 16: 'altezza e larghezza sprite
CALL setpointer(fin&, buf&, a&, l&, 0, 0)
'
loop:
a$ = INKEY$
IF a$ = "" THEN loop
CALL freemem(buf&, byte&)
LIBRARY CLOSE: WINDOW CLOSE 2: END

' DATI DELLO SPRITE-POINTER
DATA 000, 000, 000, 000
DATA 192, 000, 254, 000
DATA 252, 000, 130, 000
DATA 068, 000, 186, 000
DATA 092, 000, 162, 000
DATA 092, 000, 163, 224
DATA 071, 192, 184, 032
DATA 124, 064, 131, 160
DATA 005, 192, 250, 032
DATA 005, 192, 010, 062
DATA 004, 124, 011, 130
DATA 007, 196, 008, 058
DATA 000, 092, 015, 162
DATA 000, 092, 000, 162
DATA 000, 068, 000, 186
DATA 000, 124, 000, 130
DATA 000, 000, 000, 254
DATA 000, 000, 000, 000

END
```

Listato 1

ne e la traduzione in comode righe **Data** inseribili in qualunque listato basic.

Un **pointer-data-maker**, insomma. Torniamo dunque al nostro listato, ed ai parametri da associare alla funzione **SetPointer** della libreria Intuition.

Inutile tornare sul primo di questi parametri, come si è detto ricavabile con facilità tramite la funzione Window(7). Altrettanto semplici da definire i dati inerenti altezza, larghezza e Offset, mentre qualche parola in più è necessario spenderla sul secondo parametro, l'indirizzo ove sono memorizzati i dati dello sprite.

Intanto cominciamo col dire che questi dati, corrispondenti nel listato alla serie di valori inseriti nelle linee data, non corrispondono a quanto si è abituati a manipolare in ambiente basic. In altre parole, non è possibile adoperare "tout court" quanto prodotto, per esempio, dall'**Obj Editor** contenuto nel disco Extras, che usa un proprio formato finalizzato esclusivamente all'interprete di Amigabasic.

Nel nostro caso, la forma ed i colori dello sprite andrebbero impostati piuttosto come descritto (p. es.) nella sezione Assembly del n. 77 della rivista (pag. 66). Unica differenza, la struttura dello sprite-pointer è caratterizzata dalla presenza di una **long word** (4 byte) di valore **0** come primo ed ultimo dato (si notino gli zeri alle estremità delle linee data nel listato).

Ma l'argomento si fa piuttosto complicato, per cui soprassediamo. Chi volesse, può approfondire il concetto ripescando il n. 77 della rivista; per gli altri forniremo tra breve una soluzione di estrema facilità.

I dati dello sprite, come ovvio, dovranno essere memorizzati in un'area di memoria riservata, che non possa essere trascritta da altre applicazioni.

Per di più, il sistema hardware di gestione degli sprite richiede che questa memoria sia la cosiddetta **Chip Ram**, fisicamente identificabile nei primi **512 KB** (o nel primo Mebabyte se si dispone dei **nuovi Agnus**). Anche in questo caso, comunque, non è necessario preoccuparsi più di tanto dei dettagli: basta ricorrere alla libreria **Exec**, ed alla sua funzione **Allocmem**, penserà lei a tutto.

Nel listato, le prime istruzioni dopo l'apertura delle librerie si occupano proprio di questo particolare.

il parametro **Byte&** indica ad Allocmem quante locazioni intendiamo riservarci, mentre assegnando valore 2 (variabile **chip&**) al secondo si precisa che questi byte devono risiedere nella Chip Memory.

Dopo il richiamo della funzione Allocmem, la variabile **Buf&** conterrà l'indirizzo di inizio di quest'area, nelle cui locazioni le successive istruzioni (tramite read...data) "pokeranno" i dati dello sprite. Lo stesso indirizzo viene poi inserito come secondo parametro di Setpointer... ed il gioco è fatto.

Prima di concludere le operazioni, si ricordi che va sempre deallocata la memoria impegnata, ovvero resa di nuovo libera per altre eventuali applicazioni, tramite la funzione **Freemem**, sempre di Exec.

Unico problema rimasto insoluto è come realizzare uno sprite-pointer, e soprattutto come trasformarlo in una serie di Data come quelli presenti nel programma di queste pagine.

Ebbene, piuttosto che ammattire con calcoli e notazioni

numeriche non molto familiari, un editor a portata di mano l'abbiamo tutti: il programma Preferences. Questo, come noto, consente di modificare a proprio piacimento la forma del pointer (opzione Edit Pointer dopo aver attivato il Preferences da Workbench, oppure doppio click direttamente sull'icona Pointer), per poi salvarlo sul disco di sistema, assieme ad altre informazioni, nel file **System-configuration** contenuto nella directory **Devs**.

Ebbene, in questo file lo sprite è memorizzato esattamente nel formato che ci interessa, a partire dal byte **29**, per un totale di **32** word (1 word = 2 byte), cui sono da aggiungere le due long word azzerate (iniziale e finale). Se lo sprite è di dimensioni inferiori, lo spazio eccedente è riempito di zeri, per cui la cosa non creerà problemi di sorta.

Facile, quindi, ricavare una routine che prelevi questi (32 * 3) + 8 = **72** byte, e li trasferisca in un file Ascii come farebbe un qualunque datamaker.

E' proprio di ciò che si occupa il **listato 2**, che risulta decisivo nella personalizzazione di un pointer da basic. Il listato è abbastanza semplice, e riprende un po' la tecnica già illustrata in precedenza (vedi CCC n. **74** e **77**) per creare linee data partendo da un codice numerico puro. In sintesi, ecco i passaggi necessari per giungere al risultato desiderato:

- Effettuare una copia del disco workbench, e adoperarla per lanciare il sistema.
- Usare Preferences per disegnare un nuovo pointer, e uscire dal programma con l'opzione Save.
- Caricare il basic, e lanciare la routine di cui al listato 2, che chiameremo **PGrab**. Dopo lo start, fornire al programma l'indicazione sul drive nel quale è presente il disco con il novello pointer. Un'altra scelta riguarderà il drive ove memorizzare il file prodotto (ottimale la Ram:, dalla quale lo si può poi spostare ove si desidera, anche da Workbench). Il file prodotto, salvo vostri auspicabili interventi, avrà sempre come nome **pointer.data** (occhio a non sovrascriverlo inavvertitamente).
- Caricare il programma basic che prevede l'implementazione del nuovo pointer. A titolo di prova, si può adoperare il listato 1, dal quale andranno preventivamente eliminate tutte le righe data.
- Impartire una istruzione...

Merge "Ram: pointer.data"

...con Ram: da sostituire eventualmente con l'unità o il percorso ove è memorizzato il file (p. es. **Merge "df1: pointer.data"**).

Tutto è pronto per far lavorare il nuovo pointer. Naturalmente la stessa tecnica può essere applicata a qualunque dischetto contenga un "system-configuration" nella sua directory Devs, anche se non si è provveduto personalmente ad editare un nuovo sprite 0 (saltando i primi due passaggi, insomma).

Il che si traduce, come intuibile, nella possibilità di "prelevare" un pointer particolarmente simpatico da qualunque dischetto rispondente ai normali dettami di AmigaDos.

E tutto da basic!

```
PRINT " TRASFORMA UN POINTER DISEGNATO"
PRINT " CON IL PROGRAMMA PREFERENCES"
PRINT " IN DATA ADOPERABILI DA BASIC"

ON BREAK GOSUB fine: BREAK ON
WINDOW 2, "GRAB POINTER", (200, 70)-(370, 150), 18
COLOR 2, 3: CLS: file$ = "devs/system-configuration"
fil2$ = "pointer.data"
PRINT : PRINT " IN QUALE UNITA' SI"
PRINT " TROVA IL DISCO CON"
PRINT " IL POINTER?": PRINT : PRINT
PRINT " 0 - Drive DF0: "
PRINT " 1 - Drive DF1: "

loop:
a$ = INKEY$
IF a$ = "0" THEN file$ = "df0: "+file$
IF a$ = "1" THEN file$ = "df1: "+file$
IF a$ < > "0" AND a$ < > "1" THEN loop
CLS: PRINT: PRINT" IN QUALE UNITA'"
PRINT " MEMORIZZO"
PRINT " IL DATA-FILE?": PRINT: PRINT
PRINT " 0 - Drive DF0: "
PRINT " 1 - Drive DF1: "
PRINT " 2 - Device RAM: "

loop2:
a$ = INKEY$
IF a$ = "0" THEN fil2$ = "df0: "+fil2$
IF a$ = "1" THEN fil2$ = "df1: "+fil2$
IF a$ = "2" THEN fil2$ = "ram: "+fil2$
IF a$ < > "0" AND a$ < > "1" AND a$ < > "2" THEN loop2
CLS: PRINT: PRINT: PRINT SPC(7)"ATTENDI"

OPEN file$ FOR INPUT AS 1: null$ = INPUT$(28, 1)
pnt$ = INPUT$(72, 1): CLOSE 1
OPEN fil2$ FOR OUTPUT AS 1
FOR riga = 1 TO 72 STEP 4: PRINT #1, "DATA ";
FOR dati = 0 TO 3
a = ASC(MID$(pnt$, dati+riga, 1)): a$ = STR$(a)
IF a < 10 THEN PRINT #1, CHR$(48);
IF a < 100 THEN PRINT #1, CHR$(48);
PRINT #1, RIGHT$(a$, LEN(a$)-1);
IF dati = 3 THEN PRINT #1, "" ELSE PRINT #1, ", ";
NEXT dati
NEXT riga

CLOSE 2: CLS: PRINT: PRINT: PRINT SPC(8)"FATTO"
PRINT : PRINT" Premi un tasto"
PRINT" per finire"
loop3:
a$ = INKEY$: IF a$ = "" THEN loop3
fine:
WINDOW CLOSE 2: CLOSE: END
```

Listato 2

(S)consigli per gli acquisti

Posseggo un Commodore C/128, del quale penso di disfarmi tra un paio di anni per passare ad Amiga. In questa previsione, potreste consigliarmi un monitor che poi non debba cambiare, ed una stampante a colori a matrice di punti?

(Massimil. Gaudio - Roma)

Quando si dice un tipo previdente...

Una sola curiosità: cosa penserà di farsene, il nostro lettore, di un C/128 nei prossimi due anni?

Comunque, il problema del monitor è risolvibile con una certa facilità: il modello Commodore **1084** è compatibile con entrambi i computer, sempre che tra due anni non sia del tutto superato da altri modelli più rappresentativi (meditare, 128isti, meditare). E, comunque, diciamo che per Amiga è il modello "minimo" per cominciare (giusto cominciare, si badi bene) ad apprezzarne la grafica, anch'essa tra l'altro in fase di evoluzione.

Quanto alla stampante, in tutta sincerità la scelta di un modello ad aghi a colori non è proprio la migliore, se si considera l'alto costo dei nastri (e, soprattutto, il loro rapido degrado) accompagnato da una qualità per forza di cose non eccelsa, quantomeno rispetto al colore.

Il consiglio, insomma, sarebbe quello di ripiegare sul bianco e nero, soprattutto in considerazione della compatibilità richiesta con i due computer.

Quest'ultima esigenza, in ogni caso, limita la scelta a due modelli della Commodore, dotati tanto di interfaccia seriale Commodore che parallela centronics: la **MPS1230** (bianco e nero) e la **MPS1550C** (a colori). A meno che non si adoperi per il C/128 una interfaccia parallela esterna, in grado di "pilotare" qualunque stampante Centronics, come ad esempio la Asem. In questo caso, il problema sulla scelta del modello di stampante non sussiste più, dipende tutto solo dai gusti... e dal portafoglio.

Non funge

Molti programmi pubblicati su riviste del settore (tra cui il vostro "Supermenu Programmabile", pubblicato sul N. 77) funzionano benissimo per ciò che riguarda le istruzioni grafiche, ma presentano disallineamenti tentando di inserire messaggi sul video. Quale può essere la causa?

(Da alcune lettere)

Purtroppo non sempre gli autori dei programmi precisano in quale modalità è attivato lo schermo al momento in cui scrivono i listati. Come tutti sanno, infatti, è possibile selezionare, da Preferences del Workbench, il numero di colonne visualizzabili (**60** oppure **80**); chi dispone del televisore a colori, di solito, seleziona il valore 60, perchè meglio visibile. Chi, invece, lavora con un monitor, preferisce lo standard, più "professionale" di 80. Ne consegue che un listato scritto tenendo presente un video a 60 caratteri presenta disallineamenti se adoperato in un sistema predisposto a 80 colonne (e viceversa).

In casi come questi, comunque, è sufficiente settare diversamente il menu di Preferences prima di ricercare altre cause di malfunzionamenti. L'ideale, ovviamente, sarebbe di scriver programmi che siano in grado di stabilire, automaticamente, la modalità video, e di comportarsi di conseguenza. E', questa, un'ennesima sfida ai lettori? Può darsi...

RESET DA BASIC?

Ho letto su un libro che è possibile resettare l'Amiga da basic con una istruzione Call 16515072, che però, sul mio computer, genera sempre un errore. Inoltre, vorrei sapere a che cosa servono Getenv e Setenv.

(Peter Kormel - Legnaro)

L'istruzione basic per provocare un reset funziona, solo che va espressa in una forma sintattica diversa, assegnando prima l'indirizzo ad una variabile a 32 bit. Per esempio:

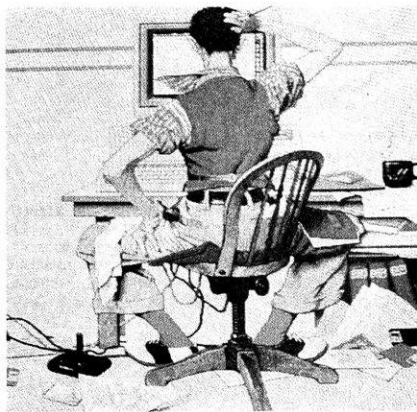
rst& = 16515072

call rst&

Dopo una breve attesa, caratterizzata da uno schermo colorato in giallo, si avrà un nuovo start di sistema.

Si tenga però presente che una chiamata diretta in Rom (in questo caso **$FC0000**) è una prassi sconsigliabile, in quanto può scontrarsi con le variazioni apportate nel tempo al sistema Amiga. Per intenderci: un'abitudine da non prendere.

Quanto a **Setenv** e **Getenv**, come chiaramente illustrato nel manuale del dos 1.3, servono rispettivamente per creare e richiamare delle variabili ambiente. Purtroppo la loro utilità pratica in ambiente Dos "ufficiale" (almeno fino alla versione 1.3) è decisamente limitata, in quanto solo il programma di lettura testi **More** e la condizione **If** possono farne uso.

Poco da aggiungere, quindi, a quel poco che il manuale descrive, in attesa di poter disporre delle chance offerte dalla versione 2.0 che, al momento in cui si scrive, sembrano limitate all'Amiga 3000.

di Ascanio Orlandini

PRONTO, CHI MODEM?

Vediamo che cosa succede quando ci colleghiamo con una banca dati e svolgiamo una "conversazione"

Chi usa per la prima volta un modem, di solito, dedica molto tempo prima di impadronirsi del "sistema" telematico. Purtroppo, oltre al tempo, è necessario spendere parecchio denaro, in termini di **scatti SIP**, dal momento che le videate delle banche dati, con cui ci colleghiamo, non sempre sono chiarissime a prima vista.

Vi proponiamo una simulazione di rice - trasmissione con la nostra banca dati, da "studiare" con calma in queste pagine prima di effettuare un vero collegameto.

Siete pronti?

Bene, accendete l'Amiga, caricate il programma di telecomunicazione (che supponiamo sia **JR-Comm**, ma un qualsiasi altro software va bene allo scopo) e... partite!

N.B: in ***corsivo e neretto*** viene descritta la digitazione compiuta dall'utente Amiga, che supponiamo parli in prima persona.

In *corsivo sottolineato*, invece, l'eventuale voce del menu selezionata. Le **schermate** che compaiono ad ogni digitazione effettuata vengono riportate in appositi riquadri, per una migliore comprensione delle stesse; anche nei riquadri viene evidenziata, in neretto e corsivo, la risposta digitata dall'utente.

Incomincio: Carico JR-Comm e richiamo il PhoneBook; clicco due volte sul nome di **BBSYSTEM** che, furbescamente, ho precedentemente memorizzato nell'apposita agenda. Automaticamente si apre la finestrella, si compone il numero di telefono e... sì! E' libero! Mi collego (vedi riquadro 1).

Purtroppo il testo della rivista non supporta... la grafica ANSI, per cui vedete solo caratteri ASCII. Per avere un'idea della grafica, rifatevi alle immagini.

Dopo aver risposto Y (oppure N) alla videata del riquadro 1, compare la schermata del riquadro 2. In questa viene indicata, come risposta, la pressione di Amiga + Z, che corrisponde all'opzione Phonebook / send Password. Nel riquadro 3 è riportata la schermata di Help, che appare invocando l'apposita richiesta (H).

Il riquadro 4 rappresenta la schermata di Area Messaggi, richiamata con A del precedente riquadro. E così via

GT POWER 15.00, 06-20-89 1:45 MST
Copyright (c) 1985, 1989 by P&M Software Co.
SYSTEMS
Se sulla riga precedente avete visto la scritta SYSTEMS colorata, rispondete con "Y" alla domanda che segue, per utilizzare la grafica ANSI. Ricordiamo che la gestione dei comandi ANSI rallenta notevolmente l'output su alcuni computer; ne consigliamo l'utilizzo soltanto a coloro che dispongono di modem ad alta velocità.
Can your terminal display ANSI graphics? *[y/n]* ***Y***

Riquadro 1

Benvenuti a Bbsystems; Sempre Un Passo Avanti
La password per i nuovi utenti è Test
Il computer che gestisce questa bbs e' un personal DM Line offerto da Digimail: soluzioni hardware e software, installazione lan (local area networks), fornitura di sistemi gestionali, desktop video e desktop publishing "chiavi in mano".
Pagamenti personalizzati con leasing e finanziamenti. Digimail s.r.l.;
Sala dimostrazioni in Via Coronelli, 10 - 20146 Milano. Tel. 02 / 426559 - 427621. Fax: 427768.
Attenzione (per i nuovi utenti):
A meno che il SysOp non vi abbia rilasciato una password di sistema dovete utilizzare TEST alla richiesta Systems password: (introdurre TEST). La vostra password verrà introdotta in seguito e sara' differente dalla System password.
Your FIRST name: ***Ascanio***
and LAST name: ***Orlandini***
Ascanio Orlandini? *[y/n]:* ***y***
You have called 38 times.
Last called on 12-09-90 at 23:58.
Password: ***(Tasto Amiga + Z)***
{n.d.R. Nota Del Red: I messaggi seguenti possono essere interrotti con Ctrl-c oppure rispondendo con un sonoro "No" alla rischiesta di scroll.}

Ecco che cosa compare dopo la digitazione del nome

Livello Utente Privilegiato
Benvenuto, Ascanio; hai a disposizione 419 minuti per effettuare le operazioni di scambio file.
Bollettini disponibili:
1. Regole generali di utilizzo.
2. Altre BBS della zona.
L. List the logon bulletins.
Q. Quit this menu.
Enter desired bulletin number: ***2***
Bollettino dedicato alla pubblicità gratuita per altre BBS:
Digital Empire di Mkv 02, 90781400
Bbs Rozzano 02, 89202186
Highland Board 0362, 901606
Spazio Monza 039, 2023478
Jack Bbs 02, 6948296
Laser Amiga 02, 27200576
Soft House Club 0373, 273188 New! 24h
Euro Elettronica 0373, 86966 Dalle 20:00 alle 8:00
Middle Earth 02, 26920027 No Telix allowed!
Chi volesse inserire altri nominativi lo può fare nell'apposita area msg. Supersalutoni, Red

Bollettini disponibili:
1. Regole generali di utilizzo.
2. Altre BBS della zona.
L. List the logon bulletins.
Q. Quit this menu.
Enter desired bulletin number: ***q***

L'elenco di altre Bbs viene costantemente aggiornato

File Section: #9 - Vari file di utilizzo generale (testi, manuali ecc.)
Message Base: 1 - Area messaggi pubblici
A) Cambio area
B) Lista bollettini
C) Cambio directory
D) Download file
E) Nuovo messaggio
F) Descrizione file
G) Logoff (fine lavoro)
H) Help
I) Nuovi file
K) Elimina messaggio
M) Messaggio al Sysop
N) Elimina "More?"
P) Chiama il Sysop
Q) Questionari
R) Lettura messaggio
RG) Lettura globale
T) Stampa un file
U) Upload file
UN) Esce dall'area
V) Versione del software W) Ricerca
X) Modo EXPERT (senza menu)
Y) Parametri correnti

n Minutes left. Enter command (? for help): ***?***

"Riassunto" delle potenzialità offerte dal s/w della Bbs

Help
A - Cambia l'area messaggi. Visualizza una lista delle aree disponibili.
B - Visualizza i bollettini disponibili.
C - Cambia la sezione file. Visualizza una lista delle sezioni disponibili.
D - Download file. Permette la ricezione di uno o più file.
E - Immissione messaggio. Può essere sia privato che pubblico.
F - Contenuto directory corrente.
G - Goodbye. Interrompe la comunicazione agganciando il telefono.
H - Help. (E' questa schermata!)

I - Cerca i file più recenti a partire da una certa data o dall'ultimo collegamento effettuato.
K - Elimina un messaggio (se l'utente è il mittente o il destinatario).
L - Lista il contenuto della directory corrente in formato DOS (occorre un adeguato livello di abilitazione).
M - Immette un messaggio privato per il Sysop.
N - Elimina/abilita il messaggio "More?" al termine di ogni schermo. Quando è disabilitato non si ferma ad ogni schermo e lo scrolling procede normalmente.
O - Esegue un programma esterno alla BBS (occorre un elevato livello di abilitazione).
P - Chiama il Sysop per una conversazione on-line.
Q - Risponde al questionario.
R - Entra nella sezione di lettura messaggi.
RG - Automaticamente percorre tutte le aree messaggi abilitate.
S - Shell al sistema operativo (occorre un livello di autorizzazione elevato).
T - Visualizza il contenuto di un file testo (come il comando Type).
U - Upload file. Consente all'utente di inviare un file.
UN - Esce da un area messaggi.
V - Visualizza la versione di GT Power correntemente in uso.
W - Cerca un file nella struttura dati della BBS.
X - Modo expert... non visualizza il menu principale, consentendo un sensibile risparmio di tempo.
Y - Visualizza le statistiche relative all'utente collegato, compresi i livelli di abilitazione.

n Minutes left. Enter command (? for help): ***a***

Videata principale contenente le possibilità di collegamento

Current Area: #1 - Area messaggi pubblici
1. Area messaggi pubblici
2. Area messaggi dal e per il Sysop
3. Domande / Risposte per la rivista Personal Computer
4. Domande / Risposte per la rivista C. C. C.
5. EMS, questa sconosciuta...
6. Parliamo un po' del C/128
7. Ma lo 0 è pari o dispari?
8. Suggerimenti per Bbsystems
9. Pubblicità per altre BBS
10. Annunci economici vari
Enter desired area number: ***4***

Reading #4 - Domande / Risposte per C. C.C.
Messages available are #1 to 32. You last read #27.
Check Kill Nonstop Enter Reply Scan Thread Quit Msg# Msg Area +, -: n
Msg Base: #4 - DOMANDE / RISPOSTE per Commodore Computer Club
Msg No: **28**. Dom. 6 - Marzo - 1990 24 : 36 (no Kill)
From: Pietro Capozzi
To: Federico Alpi
Subject: achigiovatuttociò
ALSO SEE: 26.
Ciao. Ero un vecchio 128..ista, e prima ancora un 64..ista. Ora gioco con l'Amiga 500 ed ogni tanto con l' Ms/dos. Ho un piccolo rancore con il 128 perchè non sono riuscito ad utilizzarlo come speravo. Quando sono rimasto senza, (perchè venduto), ho iniziato le crisi di astinenza ma sono passato ad un clone Ms-dos. Non soddisfatto eccomi con un Amiga 500 e nel futuro spero di riuscire a vedere il mio Amiga emulare un Ms-dos. (Chi vivrà vedrà). Tra parentesi ho avuto mille occasioni di accedere al public domine per 128 e 64 che arrivava dall'USA, ma sfortunatamente non sono mai riuscito nell'intento di scaricare qualche bit

Esempio di messaggio "lanciato" da un utente della Bbs

Msg Base: #1 - Area messaggi pubblici
Msg No: 90. Fri 7-27-90 21:03
From: Peter Stone
To: All
Subject: Protocol Modem
C'è qualcuno che sappia indicarmi quale sia il protocollo migliore da usare e le differenze tra i vari tipi disponibili? Grazie a tutti anticipatamente. Vs. neoutente Peter Stone.

Al quesito posso rispondere: comando R (replay) e...

Messages available are #1 to 92. You last read #90.
Check Kill Nonstop Enter Reply Scan Thread Quit Msg# Msg Area+,-: r
REPLY TO MESSAGE ***#90.***
From: Ascanio Orlandini
To: Peter Stone
Press Esc for new subject, or Enter to retain old subject.
Subject: Protocol Modem
Security: Private Kill protect None? ***n***
Enter message, 70 lines max., empty line to end.

1: ***Il protocollo migliore in assoluto è lo Z-modem (che ha delle funzioni***
2: ***uniche tra cui il resume: se ti cade la linea all'ultimo byte di un***
3: ***programma di 200K, che fare? Con lo Z-modem ti puoi prendere solo l'ultimo***
4: ***byte senza ri-iniziare daccapo!***
5: ***Se la linea telefonica è "pulita", allora l'Y-modem 1k è leggermente***
6: ***più rapido, ma la linea deve essere pulita!. Per non sbagliare, usa Zmodem!***
7: ***Ciao.***
Abort Continue Insert Delete Edit Topic List Read *Save*: ***s***
Storing message #93...

Ecco come rispondere ad una richiesta presente in Bbs

n Minutes left. *Enter command* (? for help): ***c***
Current Area: #9 - Vari file di utilizzo generale (testi, manuali ecc.)

Utility
1. Utility per MS-Dos
2. Utility per Amiga

Giochi
3. Giochi per MS-Dos
4. Giochi per Amiga

Tools
5. Tools per MS-Dos
6. Tools per Amiga
Programmi di comunicazione
7. Prg di comunicazione per Amiga
8. Prg di comunicazione per Ms-Dos

Varie
9. Vari file di utilizzo generale (testi, manuali ecc.)
10. Vari prg gestionali (Magazzino Fatturazione ecc...)
11. Vari file per Ms-Dos
12. Vari file per Amiga

Articoli (occorre l'abilitazione)

Uploads
13. Upload collaboratori esterni (area riservata)
14. Upload normali utenti... area libera a tutti
15. Programmi privati tra noi

Enter desired area number: ***7***

#7 - Programmi di comunicazione per Amiga selected.
File Section: #7 - Programmi di comunicazione per Amiga
n Minutes left. *Enter command* (? for help): ***f***

Area dedicata ai programmi di comunicazione per Amiga

Azcomm25.zip 32 4-16-90 Azcomm; il più diffuso software di comunicazione
Baudband.zip 57 4-16-90 Da provare...
Diga202.zip 203 4-16-90 Gestisce anche una mini-BBS
Digadocs.zip 15 4-16-90 Documentazione di DIGA
Jrcom99i.zip 156 4-16-90 Molto buono (e molto grosso)...
Ncomm18.zip 86 4-16-90 Emulaz. terminale per Amiga
Online21.zip 97 4-16-90 Come sopra..
Jrcomm1.lzh 255 9-15-90 JR-Comm v1.0 basta la parola!
Press Return to continue: ***R***

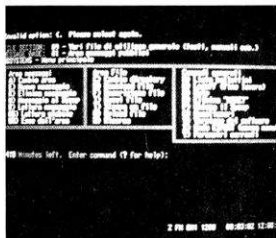
File Section: #7 - Programmi di comunicazione per Amiga

Enter desired area number: ***14***
#14 - Upload normali utenti... area libera a tutti selected.
File Section: #14 - Upload area libera a tutti

Amiga Byte 2 1 4-21-90 nocomment
Arka2 1 4-21-90 bello
Exodus!.arc 27 4-15-90 Gioco simile alla dama, amiga
Flitri_c.lzh 24 4-25-90 Filtri di testo per Amiga, con sorgenti e .doc
Lhunrc96.lzh 15 4-24-90 scompatt. lzh (100% compatibile lharc) fast (amiga)!
Lzexe.exe 13 4-20-90 Comprime .exe lasciandoli eseguibili (Msdos)
Opsvirus.lzh 13 5-02-90 Antivirus per Amiga (Lharc).
Pic1 32 4-20-90 disegno IFF 32 colori (Amiga)
Sekpac.lzh 18 4-24-90 Devpac Converter (amiga)
Subdir.exe 43 4-17-90 Utility per Harddisk Msdos
The Force 11 4-21-90 disegno iff (Amiga)
Turbo!.lzh 39 4-24-90 Imploder turbo! version the best cruncher (amiga)
Utils.zip 24 5-12-90 utility varie (Msdos - Molto Belle)
Piccino.zip 30 5-12-90 stampa esponente compresso (quickbasic 4.5)
Sruler.zip 2 5-12-90 TSR - righello (Msdos - Molto Bello)
3dpac.arc 31 5-14-90 Pacman Tridimensionale (amiga)
File Section: #14 - Upload normali utenti... area libera a tutti

Esempio di files che è possibile copiare con il modem

n Minutes left. Enter command (? for help): ***g***
Disconnect? [Y/n] ***y***
Goodbye, Ascanio Orlandini
BBSYSTEMS
Non perdete in edicola gli ultimi numeri di
Personal Computer + Commodore Computer Club
Grazie per aver chiamato BBSYSTEMS e arrivederci alla prossima volta.
Ricordiamo che il nostro numero è 02 - 57605211

Le ultime digitazioni prima di interrompere la comunicazione

QUANTO COSTA IL TUO COMMODORE

Amiga 2000 - L. 2.715.000

Microprocessore Motorola MC68000 - Clock 7.16MHz - Kickstart ROM - Memoria RAM: 1 MByte - 3 chip custom per DMA, Video, Audio, I/O - 5 Slot di Espansione Amiga Bus 100 pin Autoconfig™ - 1 Slot di Espansione 86 pin per Schede Coprocessore - 2 Slot di Espansione compatibili AT/XT - 2 Slot di Espansione compatibili XT - 2 Slot di Espansione Video - 1 Floppy Disk Drive da 3 1/2", 880 KBytes - Porta seriale RS232C - Sistema Operativo single-user, multitasking AmigaDOS - Compatibilità MS-DOS XT/AT disponibile con schede interne Janus (A2088 - A2286) - Monitor escluso

Amiga 500 - L. 995.000

Microprocessore Motorola MC68000 - Clock 7.16 MHz - Kickstart ROM - Memoria RAM: 512 KBytes - 3 Chip custom per DMA, Video, Audio, I/O - 1 Floppy Disk Driver da 3 1/2", 880 KBytes - Porta seriale RS232C - Porta parallela Centronics

Videomaster 2995 - L. 1.200.000

Desk Top Video - Sistema per elaborazini video semiprofessionale composto da genlock, digitalizzatore e alloggiamento per 3 drive A2010 - Ingressi videocomposito (2), RGB - Uscite Videocomposito, RF, RGB + sync -

Floppy Disk Driver A 1010 - L. 335.000

Floppy Disk Driver - Drive esterno da 3 1/2" - Capacità 880 KBytes - Collegabile a tutti i modelli della linea Amiga, alla scheda A2088 e al PC1

Floppy Disk Drive A 2010 - L. 280.000

Floppy Disk Drive - Drive interno aggiuntivo da 3 1/2" - Capacità 880 KBytes - Collegabile ad Amiga 2000

Hard Disk A 590 - L. 1.750.000

Hard Disk+Controller+RAM - Scheda Controller - Hard Disk da 3 1/2" 20 MBytes - 2 MBytes "fast" RAM - Collegabile all'Amiga 500

Scheda Janus A 2088 + A 2020 - L. 1.050.000

Scheda Janus XT+Floppy Disk Drive da 5 1/4", 360 KBytes - Scheda Bridgeboard per compatibilità MS-DOS (XT) in Amiga 2000 - Microprocessore Intel 8088 - Coprocessore matematico opzionale Intel 8087

A2286+A2020 - L. 1.985.000

Scheda Janus AT+Floppy Disk Drive da 5 1/4", 1.2 MBytes - Scheda Bridgeboard per compatibilità MS-DOS (AT) in Amiga 2000 - Microprocessore Intel 80287 - Clock 8 MHz - RAM: 1 MBytes on-board - Floppy Disk Controller on-board - Floppy Disk Driver disegnato per l'installazione all'interno dell'Amiga 2000 -

Scheda A2620 - L. 2.700.000

Scheda Processore Alternativo 32 bit - Scheda per 68020 e Unix - Microprocessore Motorola MC68020 - Coprocessore matematico Motorola MC68881 (opzionale MC68882)

Scheda A Unix - L. 3.250.000

Sistema Operativo AT&T Unix System V Release 3 - Per Amiga 2000 con scheda A2620 e Hard Disk 100 MBytes

Hard Disk A2092+PC5060 - L. 1.020.000

Hard Disk e controller - Hard Disk 3 1/2" ST506 - Capacità formattata 20 MBytes

Hard Disk A2090+2092 - L. 1.240.000

Hard Disk e controller - Hard Disk 3 1/2" ST506 - Capacità formattata 20 MBytes

Hard Disk A2090+A2094 - L. 1.900.000

Stesse caratteristiche del kit A2092 ma con disco da 40 MBytes

Espansione di memoria A2058 - L. 1.149.000

Espansione di memoria - Scheda di espansione per Amiga 2000 - Fornita con 2 MBytes "fast" RAM, espandibile a 4 o 8 MBytes

Scheda Video A2060 - L. 165.000

Modulatore video - Scheda modulatore video interna per Amiga 2000 - Uscite colore e monocromatica - Si inserisce nello slot video dell'Amiga 2000

Genlock Card A2301 - L. 420.000

Genlock - Scheda Genlock semiprofessionale per Amiga 2000 - Permette di miscelare immagini provenienti da una sorgente esterna con immagini provenienti dal computer

Professional Video Adapter Card A2351 - L. 1.500.000

Professional Video Adapter - Scheda Video Professionale per Amiga 2000 (B) - Genlock qualità Broadcast - Frame Grabber - Digitalizzatore - Include software di controllo per la gestione interattiva
(Disponibile da maggio '89)

A501 - L. 300.000

Espansione di memoria - Cartuccia di espansione di memoria da 512 KBytes per A500

A520 - L. 45.000

Modulatore RF - Modulatore esterno A500 - Permette di connettere qualsiasi televisore B/N o colori ad Amiga 500

A Scart - L. 27.000

Cavo di collegamento A500/A2000 con connettore per televisione SCART

Monitor a colori 1084 - L. 595.000

Monitor a colori ad alta risoluzione - Tubo 14" Black Matrix antiriflesso - Pitch 0.39 mm - Compatibile con Amiga 500/2000, PC (tutta la gamma), C64 e C128

Monitor a colori 2080 - L. 770.000

Monitor a colori ad alta risoluzione e lunga persistenza - Tubo 14" Black Matrix antiriflesso - Pitch 0.39 mm - Frequenza di raster 50 Hz - Compatibile con Amiga 500/2000, PC (tutta la gamma), C64 e C128

Monitor Monocromatico A2024 - L. 1.235.000

Monitor monocromatico a fosfori "bianco-carta" - Turbo 14" antiriflesso - (Disponibile da marzo '89)

PC60/40 - L. 7.812.000

Microprocessore Intel 80386 - Coprocessore matematico opzionale Intel 80387 - Clock 8 o 16 MHz selezionabile via software e da tastiera - Monitor monocromatico 14" - Tastiera avanzata 102 tasti con 12 funzioni - Sistema Operativo MS-DOS 3 2.1 - Interprete GW-Basic

PC60/40C - L. 8.127.000

Stessa configurazione ma con monitor 14" a colori mod. 1084

PC 60/80 - L. 10.450.000

Microprocessore Intel 80386 - Coprocessore opzionale Intel 80387 - Clock 8 o 16 MHz selezionabile via software e da tastiera - Memoria RAM: 2.5 MBytes - 1 Floppy Disk Drive da 5 1/4", 1.2 MBytes - 1 Floppy Disk Drive opzionale da 3 1/2", 1.44 MBytes - 1 Hard Disk da 80 MBytes - 2 Porte parallele Centronics - Mouse video EGA (compatibile MDA - Hercules - CGA). Emulazioni disponibili via hardware e software - Monitor monocromatico 14" - Tastiera avanzata 102 tasti con 12 tasti funzione - Sistema Operativo MS-DOS 3.21 - Ambiente Operativo Microsoft Windows/386 - Interprete GW-Basic

PC60/80C - L. 10.700.000

Stessa configurazione ma con monitor 14" a colori mod. 1084

PC40/20 - L. 4.100.000

Microprocessore Intel 80286 - Coprocessore matematico opzionale Intel 80287 - Clock 6 o 10 MHz selezionabile via software, hardware o da tastiera - Memoria RAM: 1 MByte - 1 Floppy Disk Drive da 5 1/4", 1.2 MBytes - 1 Hard Disk da 20 MBytes - Porta seriale RS232 - Porta parallela Centronics - Scheda video AGA multistandard (MDA - Hercules - CGA) Emulazioni disponibili via hardware e software - Monitor monocromatico 14" - Tastiera avanzata 102 tasti con 12 tasti funzione - Sistema Operativo MS-DOS 3.21 - Interprete GW-Basic

PC40/20C - L. 4.350.000

Stessa configurazione ma con monitor 14" a colori mod. 1084

PC 40/40 - L. 5.285.000

Microprocessore Intel 80286 - Coprocessore matematico opzionale Intel 80287 - Clock 6 o 10 MHz selezionabile via software, hardware o da tastiera - Memoria RAM: 1 MByte - 1 Floppy Disk Drive da 5 1/4", 1.2 MBytes - 1 Hard Disk da 20 MBytes - Porta seriale RS232 - Porta parallela Centronics - Scheda video AGA multistandard (MDA - Hercules - CGA) Emulazioni disponibili via hardware e software - Monitor monocromatico 14" - Tastiera avanzata 102 tasti con 12 tasti funzione - Sistema Operativo MS-DOS 3.21 - Interprete GW-Basic

PC40/40C - L. 5.535.000

Stessa configurazione ma con monitor 14" a colori mod. 1084

1352 - L. 78.000

Mouse - Collegabile con Microsoft Bus Mouse - Collegabile direttamente a PC1, PC10/20 - III, PC40 - III

PC910 - L. 355.000

Floppy Disk Drive - Drive interno aggiuntivo da 3 1/2" per PC10/20-I-II-III - Capacità 360 o 720 KBytes selezionabile tramite "config. sys" - Corredo di telaio di supporto per l'installazione in un alloggiamento per un drive da 5 1/4" - Interfaccia identica ai modelli da 5 1/4"

PC1 - L. 995.000

Microprocessore Intel 8088 - 1 Floppy Disk Drive da 5 1/4" - Porta seriale RS232C - Porta parallela Centronics - - Monitor monocromatico 12" - Tastiera 84 tasti - Sistema Operativo MS-DOS 3.2 - Interprete GW-Basic

PCEXP1 - L. 640.000

PC Expansion Box - Box esterno di espansione per PC 1 - Alimentatore aggiuntivo incluso - Contiene 3 Slot di Espansione compatibili Ibm XT - Alloggiamento per Hard Disk da 5 1/4" - Si posiziona sotto il corpo del PC1 e viene collegato tramite degli appositi connettori

PC10-III - L. 1.360.000

Microprocessore Intel 8088 Clock 4.77 MHz 9.54 MHz (double) selezionabile via software e da tastiera - Memoria RAM: 640 KBytes - 2 Floppy Disk Drive da 5 1/4", 360 KBytes - Porta seriale RS232C - Porta parallela Centronics - Porta Mouse per Mouse Commodore 1352 (compatibile Microsoft Bus Mouse - Tastiera avanzata 102 con 12 tasti funzione Sistema Operativo MS-DOS 3.21 - Interprete GW-Basic

PC10-IIIC - L. 1.675.000

Stessa configurazione ma con monitor 14" a colori mod. 1084

PC20-III - L. 2.095.000

Microprocessore Intel 8088 - Clock 4.77 MHz 9.54 MHz (double) selezionabile via software e da tastiera - 1/4", 360 KBytes - 1 Hard Disk da 20 MBytes - Porta seriale RS232C - Porta parallela Centronics - Porta Mouse per Mouse Commodore 1352 (compatibile Microsoft Bus Mouse) - Tastiera avanzata 102 con 12 tasti funzione Sistema Operativo MS-DOS 3.21 - Interprete GW-Basic

PC20-IIIC - L. 2.410.000

Stessa configurazione ma con monitor 14" a colori mod. 1084

Nuovo C64 - L. 325.000

Nuovo Personal Computer CPU 64 KBytes RAM - Vastissima biblioteca software disponibile - Porta seriale Commodore - Porta registratore per cassette - Porta parallela programmabile -

C128D - L. 895.000

Personal Computer CPU 128 KBytes RAM espandibile a 512 KBytes - ROM 48 KBytes - Basic 7.0 - Tastiera separata - Funzionante in modo 128,64 o CP/M 3.0 - Include floppy disk drive da 340 KBytes

Floppy Disk Drive 1541 II - L. 365.000

Floppy Disk Drive - Floppy Disk Drive da 5 1/4" singola faccia - Capacità 170 KBytes - Alimentazione separata - Compatibile con C64, C128, C128D

Floppy Disk Dirve 1581 - L. 420.000

Floppy Disk Drive da 3 1/2" doppia faccia - Capacità 800 KBytes - Alimentazione separata - Compatibile con C64, C128, C128D

1530 - L. 55.000

Registratore a cassette per C64, C128, C128D

Accessori per C64 - 128D

1700 - Espansione di memoria - Cartuccia di espansione di memoria a 128 KBytes per C128 **- L. 170.000**

1750 - Espansione di memoria - Cartuccia di espansione di memoria 512 KBytes per C128 - **L. 245.000**

1764 - Espansione di memoria - Cartuccia di espansione di memoria a 256 KBytes per C64 - Fornita di alimentatore surdimensionato - **L. 198.000**

16499 - Adattatore Telematico Omologato - Collegabile al C64 - Permette il collegamento a Videotel, P.G.E. e banche dati **L. 149.000**

1399 - Joystick - Joystick a microswitch con autofire - **L. 29.000**

1351 - Mouse - Mouse per C64, C128, C128D - **L. 72.000**

Monitor Monocromatico 1402 - L. 280.000

Monitor monocromatico a fosfori "bianco-carta" - Turbo 12" antiriflesso - Ingresso TTL - Compatibile con tutta la gamma PC

Monitor Monocromatico 1404 - L. 365.000

Monitor monocromatico a fosfori ambra - Turbo 14" antiriflesso a schermo piatto - Ingresso TTL - Compatibile con tutta la gamma PC - Base orientabile

Monitor Monocromatico 1450 - L. 470.000

Monitor monocromatico BI-SYNC a fosfori "bianco-carta" - Turbo 14" antiriflesso - Ingresso analogico e digitale - Doppia frequenza di sincronismo orizzontale per compatibilità con adattatori video MDA, Hercules, CGA, EGA e VGA

Monitor a colori 1802 - L. 445.000

Monitor a colori - Turbo 14" - Collegabile a C64, C128, C128D

Monitor monocromatico 1900 - L. 199.000

Monitor monocromatico a fosfori verdi - Turbo 12" antiriflesso - Ingresso videocomposito - Compatibile con tutta la gamma Commodore

Monitor a colori 1950 - L. 1.280.000

Monitor a colori BI-SYNC alta risoluzione - Turbo 14" antiriflesso - Ingresso analogico e digitale - Doppia frequenza di sincronismo orizzontale per compatibilità con adattatori video MDA, Hercules, CGA, EGA e VGA

Stampante MPS 1230 - L. 465.000

Stampante a matrice di punti - Testina a 9 aghi - 120 cps - Bidirezionale - 80 colonne - Near Letter Quality - Stampa grafica - Fogli singoli e modulo continuo - Trascinamento a trattore e/o frizione - Interfaccia seriale Commodore e parallela Centronics - Compatibile con tutti i prodotti Commodore

MPS 1230R - L. 19.000

Nastro per stampante

Stampante MPS 1500C - L. 495.000

Stampante a colori a matrice di punti - Testina a 9 aghi - 130 cps - Bidirezionale - 80 colonne - Supporta nastro a colori o nero - Near Letter Quality - Stampa grafica - Fogli singoli e modulo continuo - Trascinamento a trattore e/o frizione - Interfaccia parallela Centronics - Compatibile con la gamma Amiga e PC

MPS1500R - L. 37.000

Nastro a colori per stampante

MPS1500R - L.37.000

Nastro a colori per stampante

Stampante MPS 1550C - L. 575.000

Stampante a colori a matrice di punti - Testina a 9 aghi - 130 cps - Bidirezionale - 80 colonne - Supporta nastro a colori o nero - Near Letter Quality - Stampa grafica - Fogli singoli e modulo continuo - Trascinamento a trattore e/o frizione - Interfaccia seriale Commodore e parallela Centronics - Compatibile con tutti i prodotti Commodore

I COMMODORE POINT

LOMBARDIA

Milano
• AL RISPARMIO - V.LE MONZA 204
• BCS - VIA MONTEGANI 11
• BRAHA A. - VIA PIER CAPPONI 5
• E.D.S. - C.SO PORTA TICINESE 4
• FAREF - VIA A. VOLTA 21
• FLOPPERIA - V.LE MONTENERO 31
• GBC - VIA CANTONI 7 - VIA PETRELLA 6
• GIGLIONI - V.LE LUIGI STURZO 45
• L'UFFICIO 2000 - VIA RIPAMONTI 213
• LOGITEK - VIA GOLGI 60
• LU - MEN - VIA SANTA MONICA 3
• MARCUCCI - VIA F.LLI BRONZETTI 37
• MELCHIONI - VIA P. COLLETTA 37
• MESSAGGERIE MUSICALI - GALLERIA DEL CORSO 2
• NEWEL - VIA MAC MAHON 75
• PANCOMMERZ ITALIA - VIA PADOVA 1
• SUPERGAMES - VIA VITRUVIO 38
• 68000 E DINTORNI - VIA WASHINGTON 91

Provincia di Milano
• GINO FERRARI CENTRO HI-FI - VIA MADRE CABRINI 44 - S. ANG. LODIGIANO
• F.LLI GALIMBERTI - VIA NAZIONALE DEI GIOVI 28/36 - BARLASSINA
• TECNOLUX - VIA PIETRO NENNI 5 - BERNATE TICINO
• OGGIONI & C. - VIA DANTE CESANA 27 - CARATE BRIANZA
• AL RISPARMIO - VIA U. GIORDANO 57 - CINISELLO BALSAMO
• GBC - V.LE MATTEOTTI 66 - CINISELLO BALSAMO
• CASA DELLA MUSICA - VIA INDIPENDENZA 21 - COLOGNO MONZESE
• PENATI - VIA VERDI 28/30 - CORBETTA
• EPM SYSTEM - V.LE ITALIA 12 - CORSICO
• P.G. OSTELLARI - VIA MILANO 300 - DESIO
• CENTRO COMPUTER PANDOLFI - VIA CORRIDONI 18 - LEGNANO
• COMPUTEAM - VIA VECELLIO 41 - LISSONE
• M.B.M. - C.SO ROMA 112 - LODI
• L'AMICO DEL COMPUTER - VIA CASTELLINI 27 - MELEGNANO
• BIT 84 - VIA ITALIA 4 - MONZA
• IL CURSORE - VIA CAMPO DEI FIORI 35 - NOVATE MIL.
• I.C.O. - VIA DEI TIGLI 14 - OPERA
• R & C ELGRA - VIA SAN MARTINO 13 - PALAZZOLO MIL.
• ESSEGIEMME SISTEMI SAS - VIA DE AMICIS 24 - RHO
• TECNO - CENTRO - VIA BARACCA 2 - SEREGNO
• NIWA HARD&SOFT - VIA B. BUOZZI 94 - SESTO SAN GIOV.
• COMPUTER SHOP - VIA CONFALONIERI 35 - VILLASANTA
• ACTE - VIA B. CREMIGNANI 13 - VIMERCATE
• IL COMPUTER SERVICE SHOP - VIA PADANA SUPERIORE 197 - VIMODRONE

Bergamo
•D.R.B. - VIA BORGO PALAZZO 65
• TINTORI ENRICO &C. - VIA BROSETA 1
• VIDEO IMMAGINE - VIA CARDUCCI c/o CITTA' DI MERCATO

Provincia di Bergamo
BERTULEZZI GIOVANNI - VIA FANTONI 48 - ALZANO LOMBARDO
• COMPUTER SHOP - VIA VITTORIO VENETO 9 - CAPRIATE SAN GERVASIO
• B M R - VIA BUTTARO 4/T - DALMINE
• MEGABYTE 2 - VIA ROMA 61/A - GRUMELLO
• OTTICO OPTOMETRISTA ROVETTA - P.ZZA GARIBALDI 6 - LOVERE
• COMPUTER POINT - VIA LANTIERI 52 - SARNICO
• A.B. INFORMATICA - STRADA STATALE CREMASCA 66 - URGNANO

Brescia
• MASTER INFORMATICA - VIA F.LLI UGONI 10/B

PROVINCIA DI BRESCIA
• MISTER BIT - VIA MAZZINI 70 - BRENO
• CAVALLI PIETRO - VIA 10 GIORNATE 14 BIS - CASTREZZATO
• VIETTI GIUSEPPE - VIA MILANO 1/B - CHIARI
• MEGABYTE - P.ZZA MALUEZZI 14 - DESENZANO DEL GARDA
• BARESI RINO &C. - VIA XX SETTEMBRE 7 - GHEDI
• INFO CAM - VIA PROVINCIALE 3 - GRATACASOLO
• "PAC-LAND" di GARDONI - CENTRO COM.le - LA CASA DI MARGHERITA D'ESTE - VIA GIORGIONI 21

Como
• IL COMPUTER - VIA INDIPENDENZA 90
• 2M ELETTRONICA - VIA SACCO 3

Provincia di Como
• ELTRON - VIA IV NOVEMBRE 1 - BARZANO'
• DATA FOUND - VIA A. VOLTA 4 -ERBA
• CIMA ELETTRONICA - VIA L. DA VINCI 7 - LECCO
• FUMAGALLI - VIA CAIROLI 48 -LECCO
• RIGHI ELETTRONICA - VIA G. LEOPARDI 26 - OLGIATE COMASCO

Cremona
• MONDO COMPUTER - VIA GIUSEPPINA 11/B
• PRISMA - VIA BUOSO DA DOVARA 8
• TELCO - P.ZZA MARCONI 2/A

Provincia di Cremona
• ELCOM - VIA IV NOVEMBRE 56/58 - CREMA
• EUROELETTRONICA - VIA XX SETTEMBRE 92/A - CREMA

Mantova
• COMPUTER CANOSSA - GAL. FERRI 7
• 32 BIT - VIA C. BATTISTI 14
• ELET. di BASSO - V.LE RISORGIMENTO 69

Provincia di Mantova
• CLICK - ON COMPUTER - S.S. GOITESE 168 - GOITO

Pavia
• POLIWARE - C.SO C. ALBERTO 76
• SENNA GIANFRANCO - VIA CALCHI 5

Provincia di Pavia
• A. FERRARI - C.SO CAVOUR 57 - MORTARA
• LOGICA MAINT - V.LE M.TE GRAPPA 32 - VIGEVANO
• M. VISENTIN - C.SO V. EMANUELE 76 - VIGEVANO

Sondrio
• CIPOLLA MAURO - VIA TREMOGGE 25

Provincia di Sondrio
• FOTONOVA - VIA VALERIANA 1 - S.PIETRO DI BERBENNO

Varese
• ELLE - EFFE - VIA GOLDONI 35
• IL C.TRO ELET. - VIA MORAZZONE 2
• SUPERGAMES - VIA CARROBBIO 13

Provincia di Varese
• BUSTO BIT - VIA GAVINANA 17-BUSTO A.
• MASTER PIX - VIA S.MICHELE 3 - BUSTO A.
• PUNTO UFFICIO - VIA R.SANZIO 8 - GALLARATE
• GRANDI MAGAZZINI BOSSI - VIA CLERICI 196 - GERENZANO
• J.A.C. - C.so MATTEOTTI 38 - SESTO C.

PIEMONTE

Alessandria
• BIT MICRO - VIA MAZZINI 102
• SERV. INFOR. - VIA ALESSANDRO III 47

Provincia di Alessandria
• SONY ITALIANA - VIA G. MANARA 7 - CASALE MONFERRATO
• SGE ELETTRONICA - VIA BANDELLO 19 - TORTONA
• COMPUTER TEMPLE - VIA F. CAVALLOTTI 13 - VALENZA

Asti
• ASTI GAMES - C.SO ALFIERI 26
• RECORD - C.SO ALFIERI 166/3 (Galleria Argenta)

Cuneo
• ROSSI COMPUTERS - C.SO NIZZA 42

Provincia di Cuneo
• PUNTO BIT - C.SO LANGHE 26/C - ALBA
• BOSETTI - VIA ROMA 149 - FOSSANO
• COMPUTERLAND - VIA MAZZINI 30/32 - SALUZZO

Novara
• PROGRAMMA 3 - V.LE BUONARROTI 8
• PUNTO VIDEO - C.so RISORGIMENTO 39/B

Provincia di Novara
• COMPUTER - VIA MONTE ZEDA 4 -ARONA
• ALL COMPUTER - C.SO GARIBALDI 106 - BORGOMANERO
• S.P.A. - C.SO DISSEGNA 21/BIS - DOMODOSSOLA
• ELLIOTT COMPUTER SHOP - VIA DON MINZONI 32 - INTRA
• TRISCONI VALERIA - VIA MAZZINI 90 - OMEGNA

Torino
• ABA ELETTRONICA - VIA C. FOSSATI 5/P
• ALEX COMPUTER E GIOCHI - C.SO FRANCIA 333/4
• COMPUTER HOME - VIA SAN DONATO 46/D
• COMPUTING NEW - VIA M. POLO 40/E
• C.D.M. ELETTR. - VIA MAROCHETTI 17
• DE BUG - C.SO V. EMANUELE II 22
• DESME UNIVERSAL - VIA S.SECONDO 95
• FDS ALTERIO - VIA BORGARO 86/D
• IL COMPUTER - VIA N. FABRIZI 126
• MICRONTEL - C.SO D. degli ABRUZZI 28
• PLAY GAMES SHOP - VIA C. ALBERTO 39/E
• RADIO TV MIRAFIORI - C.SO UNIONE SOVIETICA 381
• SMT ELETTRONICA - VIA BIBIANA 83/bis

Provincia di Torino
• PAUL E CHICO VIDEOSOUND - VIA V.EMANUELE 52 - CHIERI
• BIT INFORMATICA - VIA V. EMANUELE 154 - CIRIE'
• HI - FI CLUB - C.SO FRANCIA 92C - COLLEGNO
• MISTER PERSONAL - VIA CATTANEO 52 - FAVRIA
• I.C.S. - VIA TORINO 73 - IVREA
• DAG - VIA I MAGGIO 40 - LUSERNA S. GIOVANNI
• EUREX - C.SO INDIPENDENZA 5 - RIVAROLO CANAVESE
• DIAM INFORMATICA - C.SO FRANCIA 146 bis - RIVOLI
• FULLINFORMATICA - VIA V.VENETO 25 - RIVOLI
• GAMMA COMPUTER - VIA CAVOUR 3A-3B - SET.TORINESE

Vercelli
•ELETTROGAMMA - C.SO BORMIDA 27 ang. V.Montanara
• ELETTRONICA - STRADA TORINO 15

Provincia di Vercelli
• C.S.I. TEOREMA - VIA LOSANA 9 - BIELLA
• SIGEST - VIA BERTODANO 8 - BIELLA
• REMONDINO FRANCO - VIA ROMA 5 - BORGOSESIA
• FOTOSTUDIO TREVISAN - VIA XXV APRILE 24/B - COSSATO
• STUDIO FOTOGRAFICO IMARISIO - P.ZZA M. LIBERTA' 7 - TRINO

VENETO

Belluno
• UP TO DATE - VIA V. VENETO 43

Provincia di Belluno
• GUERRA COMPUTERS - V.LE MAZZINI 10/A - FELTRE

Padova
• BIT SHOP - VIA CAIROLI 11
• COMPUMANIA - VIA T. CAMPOSANPIERO 37
• D.P.R. DE PRATO R. - V.LO LOMBARDO 4
• G.F. MARCATO - VIA MADONNA DELLA SALUTE 51/53
• SARTO COMPUTER - VIA ARMISTIZIO 79

Provincia di Padova
• COMPUTER SERVICE - BORGO TREVISO 150 - CITTADELLA

Treviso
• BIT 2000 - VIA BRANDOLINI D'ADDA 14
• GUERRA EGIDIO & C. - V.LE CAIROLI 95

Provincia di Treviso
• DE MARIN COMPUTERS - VIA MATTEOTTI 142 - CONEGLIANO
• SIDESTREET - VIA SALVO D'ACQUISTO 8 - MONTEBELLUNA
• FALCON ELETTROAUDIOVIDEO - VIA TERRAGGIO 116 - PREGANZIOL

Venezia
• GUERRA EGIDIO & C. - VIA BISSUOLA 20/A - MESTRE
• TELERADIO FUGA - SAN MARCO 3457

Provincia di Venezia
• GUERRA EGIDIO & C. - VIA VIZZOTTO 29 - SAN DONA' DI PIAVE
• REBEL - VIA F. CRISPI 10 - SAN DONA' DI PIAVE

Verona
• CASA DELLA RADIO - VIA CAIROLI 10
• TELESAT - VIA VASCO DE GAMA 8

Provincia di Verona
• UBER - CP 0363(RAG.SOC. DERTA) - VIA MASCAGNI 31 -CASTEL D'AZZANO
• FERRARIN - VIA DEI MASSARI 10 - LEGNAGO
• COMPUTERS CENTER - VIA CANTORE 26 - VILLAFRANCA

Vicenza
• ELET. BISELLO - V.LE TRIESTE 427/429
• SCALCHI MARKET - VIA CA' BALBI 139

Provincia di Vicenza
• SCHIAVOTTO - VIA ZANELLA 21 - CAVAZZALE
• GUERRA E. & C. - V.LE DELLE INDUSTRIE - MONTECCHIO MAGGIORE

FRIULI VENEZIA GIULIA

Gorizia
• E.CO. ELETTRONICA - VIA F.LLI COSSAR 23

Trieste
• AVANZO GIACOMO - P.ZZA CAVANA 7
• COMPUTER SHOP - VIA P. RETI 6
• COMPUTIGI - VIA XX SETTEMBRE 51
• CTI - VIA PASCOLI 4

Udine
• MOFERT 2 - VIA LEOPARDI 21
• R.T. SISTEM UDINE - VIA L. DA VINCI 99

Provincia di Udine
• IL PUNTO ELETTRONICO - VIA VENDRAMIN 184 - LATISANA
• IDRENO MATTIUSSI &C. - VIA LICINIANA 58 - MARTIGNACCO

TRENTINO ALTO ADIGE

Bolzano
• COMPUTER POINT - VIA ROMA 82/A
• MATTEUCCI PRESTIGE - VIA MUSEO 54

Provincia di Bolzano
• RADIO MAIR-ELECTRO - VIA CENTRALE 70 - BRUNICO
• ELECTRO RADIO HENDRICH - VIA DELLE CORSE 106 - MERANO
• ERICH KONTSCHIEDER - PORTICI 313 - MERANO
• ELECTRO TAPPEINER - P.ZZA PRINCIPALE 90 - SILANDRO

Trent0
• CRONST - VIA G. GALILEI 25

Provincia di Trento

• AL RISPARMIO - C.SO VERONA 138 - ROVERETO

LIGURIA
Genova
• ABM COMPUTER - P.ZZA DE FERRARI 24 rosso
• CAPRIOTTI G. - IA MAMIANI 4r - SAMPIERDARENA
• C.tro ELET. - VIA CHIARAVAGNA 10 R - VIA SESTRI 69R
• COM.le SOTTORIPA - VIA SOTTORIPA 115/117
• FOTOMONDIAL - VIA DEL CAMPO 3-5-9-11-13 r
• LA NASCENTE - VIA SAN LUCA 4/1
• PLAY TIME - VIA GRAMSCI 3/5/7 rosso
• RAPPR-EL - VIA BORGORATTI 23 R
Imperia
• CASTELLINO - VIA BELGRANO 44
Provincia di Imperia
• CENTRO HI-FI VIDEO - VIA DELLA REPUBBLICA 38 -SANREMO
• CASTELLINO - VIA GENOVA 48 - VENTIMIGLIA
La Spezia
• I.L. ELETTRONICA - VIA V. VENETO 123
Provincia di La Spezia
• I.L. ELETTRONICA - VIA AURELIA 299 - FORNOLA DI VEZZANO
Savona
• CASTELLINO - C.SO TARDY E BENECH 101
Provincia di Savona
• CELESIA ENZA - VIA GARIBALDI 144 - LOANO

EMILIA
Bologna
• EUROELETTRICA - VIA RANZANI 13/2
• MINNELLA ALTA FEDELTA' - VIA MAZZINI 146/2
• MORINI & FEDERICI - VIA MARCONI 28/C
• STERLINO - VIA MURRI 73/75
Provincia di Bologna
• S.C. COMPUTERS - VIA E. FERMI 4 - CASTEL SAN PIETRO
• S.P.E. INFORMATICA - VIA DI MEZZO PONENTE 385 - CREVALCORE
• ARCHIMEDE SISTEMI - VIA EMILIA 124 - S. LAZZARO DI SAVENA
Modena
• CO - EL - VIA CESARI 7
• ORSA MAGGIORE - P.ZZA MATTEOTTI 20
• VIDEO VAL WILLY COMPUTERS - VIA CANALETTO 223
Provincia di Modena
• NEW MEDIA SYSTEM - VIA ROMA 281 - SOLIERA
Parma
• BABARELLI G. - VIA B. PARENTE 14/A/B
Provincia di Parma
• PONGOLINI - VIA CAVOUR 32 - FIDENZA
Piacenza
• COMPUTER LINE - VIA G. CARDUCCI 4
• DELTA COMPUTER - VIA M. DELLA RESISTENZA 15/G

TEGGIO EMILIA
• COMPUTERLINE - VIA SAN ROCCO 10/C
• POOL SHOP - VIA EMILIA S. STEFANO 9/C
Provincia di Reggio Emilia
• MACCHIONI - VIA STATALE 467 - CASALGRANDE
ROMAGNA
Ferrara
• BUSINESS POINT - VIA CARLO MAYER 85
Forlì
• COMPUTER VIDEO CENTER - VIA CAMPO DI MARTE 122
Provincia di Forlì
• TOP BIT - VIA VENETO 12 - FORLIMPOPOLI
• COMPUTER HOUSE - V.LE TRIPOLI 193/D - RIMINI
• EASY COMPUTER - VIA LAGOMAGGIO 50 - RIMINI

REPUBBLICA S. MARINO
Ravenna
• COMPUTER HOUSE - VIA TRIESTE 134
Provincia di Ravenna
• ARGNANI - P.ZZA DELLA LIBERTA' 5/A - FAENZA
• ELECTRON INFORMATICA - VIA F.LLI CORTESI 17 - LUGO
• P.L.Z. INFORMATICA - P.ZZA SERCOGNANI 6 - FAENZA

TOSCANA
Arezzo
• DELTA SYSTEM - VIA PIAVE 13
Firenze
• ATEMA - VIA BENEDETTO MARCELLO 1a-1b
• ELETTRONICA CENTOSTELLE - VIA CENTO STELLE 5/a-b
• HELP COMPUTER - VIA DEGLI ARTISTI 15-A
• TELEINFORMATICA TOSCANA -VIA BRONZINO 36
Provincia di Firenze
• WAR GAMES - VIA R. SANZIO 126/A - EMPOLI
• NEW EVM COMPUTER - VIA DEGLI INNOCENTI 2 - FIGLINE VALDARNO
• C.tro INFOR. - VIA ZNOJMO 41 - PONTASSIEVE
• COSCI F.LLI - VIA ROMA 26 - PRATO
• BARBAGLI C. ELET. - VIA F. BONI 80 - PRATO
Grosseto
• COMPUTER SERVICE - VIA DELL'UNIONE 7
Livorno
• ETA BETA - VIA SAN FRANCESCO 30
• FUTURA 2 - VIA CAMBINI 19
Provincia di Livorno
• PUNTO ROSSO - VIA BARONTINI 28 - PIOMBINO
Provincia di Lucca
• IL COMPUTER - V.LE COLOMBO 216 - LIDO DI CAMAIORE
• SANTI VITTORIO - VIA ROMA 23 - S. ROMANO GARFAGNANA
• TOP GAMES - VIA S. ANDREA 122 - VIAREGGIO
Massa
• EURO COMPUTER - P.ZZA G. BERTAGNINI 4
Carrara
• RADIO LUCONI - VIA ROMA 24/B
Pisa
• ELECTRONIC SERVICE - VIA DELLA VECCHIA TRANVIA 10
• PUCCINI S.- CP 1199 (RAG.SOC. MAREX) - VIA C.CAMMEO 64
• TONY HI-FI - VIA CARDUCCI
Provincia di Pisa
• M.C. INFORMATICA - VIA DEL CHIESINO 4 - PONTEDERA (PI)
Pistoia
• ELECTRONIC SHOP - VIA DEGLI SCALZI 3
Provincia di Pistoia
• ZANNI &C. - C.SO ROMA 45 - MONTECATINI T.

Siena
• R. BROGI - P.ZZA GRAMSCI 28
• VIDEO MOVIE - VIA GARIBALDI 17
Provincia di Siena
• ELETTRONICA di BIFOLCHI - VIA DI GRACCIANO NEL CORSO 111 - MONTEPULCIANO

LAZIO
• CENTRO INF. - D.R.R. srl - TEL. 06-5565672

UMBRIA
Perugia
• MIGLIORATI - VIA S. ERCOLANO 3-10
Provincia di Perugia
• COMPUTER STUDIO'S - VIA IV NOVEMBRE 18/A - BASTIA UMBRA
• WARE - VIA DEI CASCERI 31 - CITTA'DI CASTELLO
Terni
• CGS SOFTWARE HOUSE - VIA DONIZETTI 71/A

BASILICATA
Matera
• G. GAUDIANO ELECTRONICS - VIA ROMA ang. XX SETTEMBRE 1

PUGLIA
Bari
• ARTEL - VIA GUIDO D'ORSO 9
• COMPUTER'S ARTS - V.LE MEUCCI 12/B
• PAULICELLI S. & F. - VIA FANELLI 231/C
Provincia di Bari
• F. FAGGELLA - C.SO GARIBALDI 15 - BARLETTA
• G.FAGGELLA - P.ZZA D'ARAGONA 62A - BARLETTA
• LONUZZO G. - VIA NIZZA 21 - CASTELLANA
• TECNOUFF. - VIA RICASOLI 54 - MONOPOLI
• TANGORRA N. - C.SO V.EMANUELE 130/B - TRIGGIANO
Brindisi
• MARANGI E NICCOLI - VIA PROV. SAN VITO 165
Provincia di Brindisi
• MILONE G. - VIA S.F. D'ASSISI 219 - FRANCAVILLA FONTANA
Foggia
• BOTTICELLI G. - VIA SAV POLLICE 2
• E.C.I. COMPUTER - VIA ISONZO 28
• LA TORRE - V.LE MICHELANGELO 185
Provincia di Foggia
• IL DISCOBOLO - VIA T. SOLIS 15 - SAN SEVERO
Lecce
• BIT - VIA 95 REGG.NTO FANTERIA 87/89
Provincia di Lecce
• TECNO UFFICIO - P.ZZA GIOVANNI XXIII 10 - GALLIPOLI
• CEDOK INFORMATICA - VIA UMBERTO I 116 - TRICASE
Taranto
• ELETTROJOLLY C.tro - VIA DE CESARE 13
• TEA - TEC. ELET. AV. - VIA R. ELENA 101

CAMPANIA
Provincia di Avellino
• FLIP FLOP - VIA APPIA 68 - ATRIPALDA
Benevento
• E.CO. INF. - VIA PEPICELLI 21/25
Caserta
• ENTRY POINT - VIA COLOMBO 31
• O.P.C. - VIA G. M. BOSCO 24
Provincia di Caserta
• M.P. COMPUTER - VIA NAPOLI 30 - MADDALONI
• DAMIANO - C.SO V. EMANUELE 23 -ORTA DI ATELLA
• FUSCO B. - VIA NAPOLI 24 - VAIRANO PATERNORA (FRAZ. VAIRANO SCALO)
• LINEA CONTABILE - VIA OSPEDALE 72/76 - SESSA A. (CE)
Napoli
• BABY TOYS - VIA CISTERNA DELL'OLIO 5/BIS
• CASA MUSICALE RUGGIERO - P.ZZA GARIBALDI 74 (INT. STAZ. F.F. S.S.)
• C.tro ELET. CAMPANO - VIA EPOMEO 121
• CI.AN - GALLERIA VANVITELLI 32
• CINE NAPOLI - VIA S. LUCIA 93/95
• DARVIN - CALATA SAN MARCO 26
• GIANCAR 2 - P.ZZA GARIBALDI 37
• ODORINO - L.GO LALA 22 A-B
• R 2 - VIA F. CILEA 285
• SAGMAR - VIA S. LUCIA 140
• TOP VIDEO - TOP COMPUTER - VIA S. ANNA DEI LOMBARDI 12
• VIDEOFOTOMARKET - VIA S. BRIGIDA 19
Provincia di Napoli
• ELECTRONIC DAY - VIA DELLE PUGLIE 17 - CASORIA
• TUFANO - S.S. SANNITICA 87 KM 7 - CASORIA
• SOF SUD - V.LE EUROPA 59 - CASTEL/MARE DI STABIA
• ELETTRONICA 2000 - C.SO DURANTE 40 - FRATTAMAGGIORE
• SPADARO - VIA ROMANI 93 - MADONNA DELL'ARCO
• GATEWAY - VIA NAPOLI 68 - MUGNANO
• VISPINI & DI VUOLO - VIA A.ROSSI 4 - POMPEI
• SPY CASH & CARRY - P.ZZA ARENELLA 6/A - NAPOLI
• NUOVA INFORMATICA SHOP - VIA LIBERTA' 185/191 - PORTICI
• BASIC COMPUTER - C.SO GARIBALDI 34 - POZZUOLI
• V.C. - C.SO SECONDIGLIANO 562/B - SECONDIGLIANO
• F. ELETTRONICA - VIA SARNO 102 - STRIANO
• TECNO - VIA V. VENETO 48 - TORRE DEL GRECO
Salerno
• COMPUMARKET - VIA BELVEDERE 35
• COMPUTER MARKET - C.SO VITTORIO EMANUELE 23
Provincia di Salerno
• KING COMPUTER - VIA OLEVANO 56 - BATTIPAGLIA
• DIMER POINT - V.LE AMENDOLA 36 - EBOLI
• IACUZIO F. - VIA MUNICIPIO 14 - MERCATO SAN SEVERINO
• COMPUTER SERVICE - VIA L.DA VINCI 81 - SCAFATI

CALABRIA
Catanzaro
• C. & G. COMPUTER - VIA F. ACRI 28
• PAONE S. & F. - VIA F. ACRI 93/99
Provincia di Catanzaro
• COMPUTER HOUSE - VIA BOLOGNA (L.GO OSPEDALE) - CROTONE
• RIOLO F.LLI - VIA VENEZIA 1/7 - CROTONE
• ING. FUSTO S. - C.SO NICOTERA 99 - LAMEZIA TERME
Cosenza
• MAISON DE L'INFORMATIQUE - VIA PASQUALE ROSSI 34/C
• SIRANGELO COMP. - VIA N. PARISIO 25
Provincia di Cosenza
• HI-FI ALFANO G. - VIA BALDACCHINI 109 - AMANTIA
• ELIGIO ANNICCHIARICO &C. - VIA ROMA 21 - CASTROVILLARI
• ALFA COMPUTER - VIA NAZIONALE 341/A - CORIGLIANO SCALO
REGGIO CALABRIA
• CONTROL SYSTEM - VIA S.F DA PAOLA 49 D
• SYSTEM HOU. - VIA FIUME ang. PALESTINO 1
Provincia di Reggio Calabria
• COMPUTER SHOP - V.LE MATTEOTTI 36/38 - LOCRI
• PICIEFFE - C.SO F. S. ALESSIO 19 - TAURIANOVA
SICILIA
• CENTRO INF. - ITALSOFT SRL - TEL. 0935-696090

DIMENSIONE
AVVENTURA
JONATHAN
OGNI MESE
IN EDICOLA
JONATHAN
DIMENSIONE AVVENTURA
TANZANIA
Safari
in mongolfiera
INDIA
I tesori
di Golconda
MONTAGNA
Trilogia
messicana
SPECIALE
Avventura
nei cieli